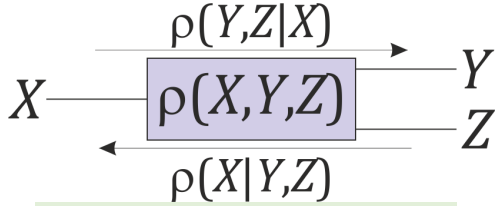
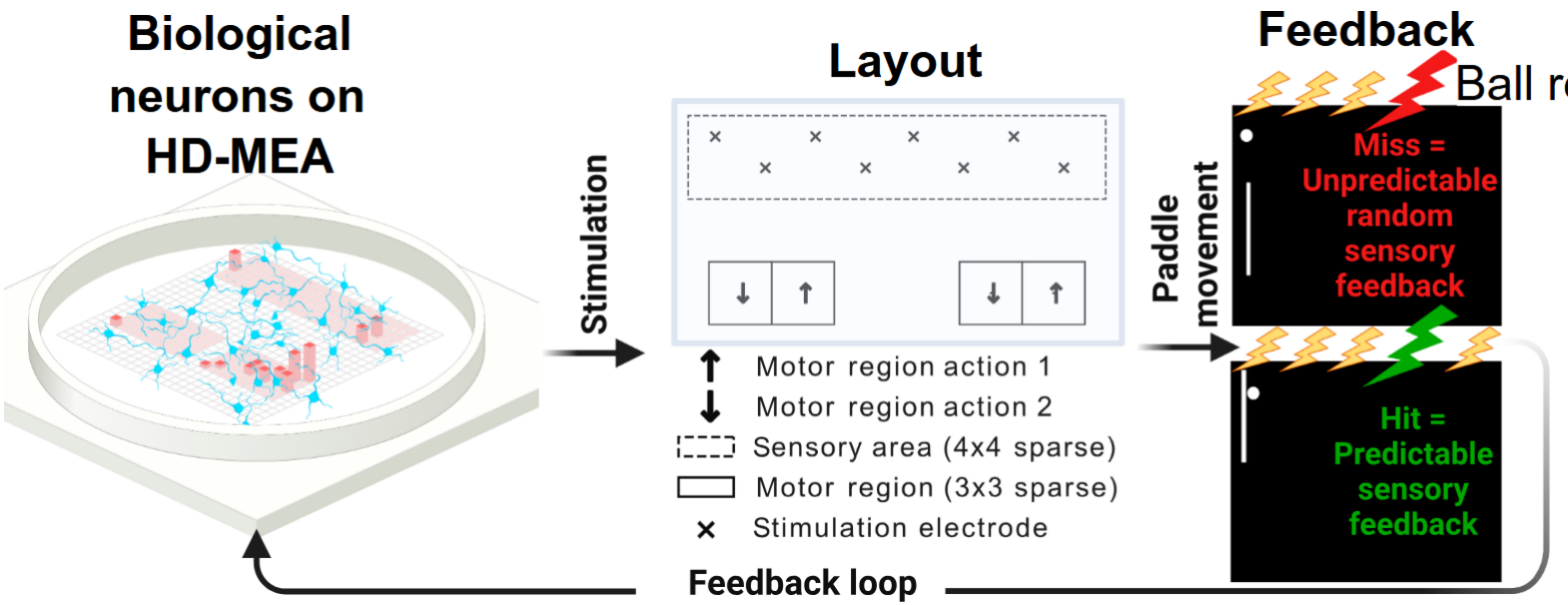
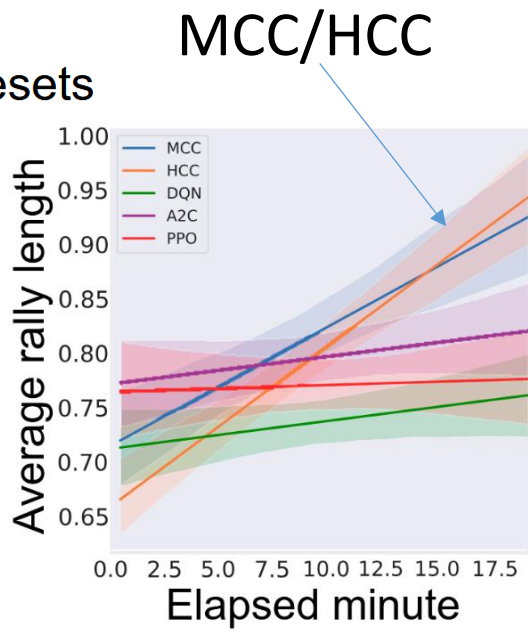


Biology-inspired neural networks with multi-directional propagation of values and distributions

Science: 10.34133/cbsystems.0336 (2025): ~ 10^6 of **biological neurons** trained to play **Pong**, **Doom** win with modern Reinforcement Learning





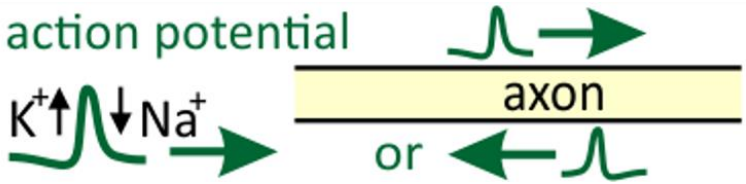
Currently **MLP: multi-layer perceptron, KAN: Kolmogorov-Arnold Networks**

Let's **extend** them toward **mathematical properties of biological neurons**

"Rats were found to prefer situations involving certainty to (...) uncertainty"

In practice world is probabilistic (modelling?)

Biological axon: bidirectional propagation



2WNN (two-way), **joint distribution neurons**

no Bayes: $\rho(Y,Z|X)$, X, $\rho(X,Y,Z)$, Y, Z, $\rho(X|Y,Z)$

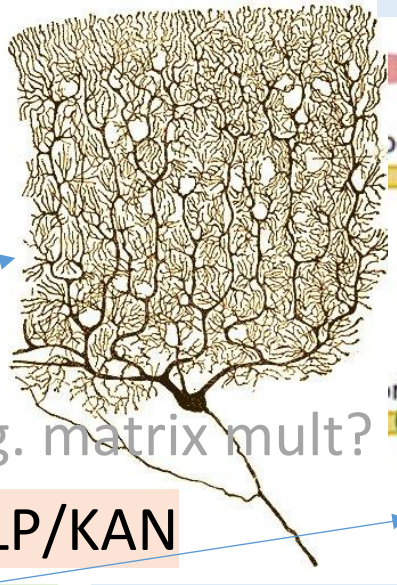
Biology-inspired neural networks with multi-directional propagation of values and distributions

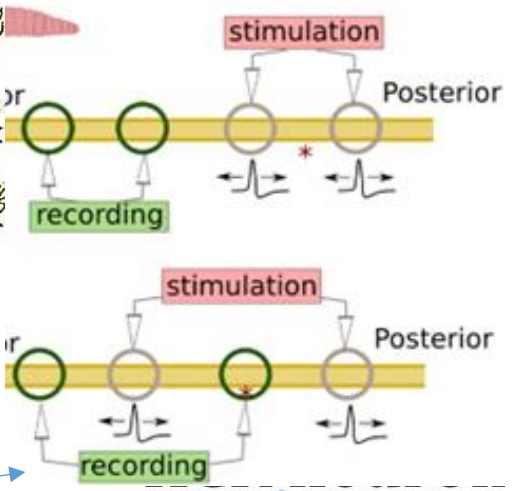
Biological NNs still superior than ANNs

increasing # of neurons is not enough

Too complex to get with spiking NN?

e.g. **bursts**, **spike annihilation**, synchronize

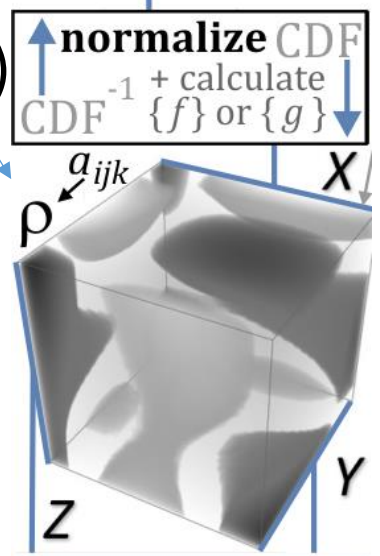




Extract mathematics to **logical neuron**? E.g. matrix mult?

what properties for BNN superiority? vs MLP/KAN

- Biological axon has **two-way propagation** (from joint dist.?)
- E.g. rats use **uncertainty** (propagate (exp. val., **variance**)?)
- biologically plausible **training** (information bottleneck?)



Upgrades of KAN arXiv:2404.19756, Tegmark, >5 citations/day

HCRNN – week later arXiv:2405.05097 from arXiv:1804.06218 adding to KAN: **two-way propagation**, of values and **distribut.**, pairwise to **higher dependencies**, **additional training ways**

direct, **IB**, **tensor decomposition** ...

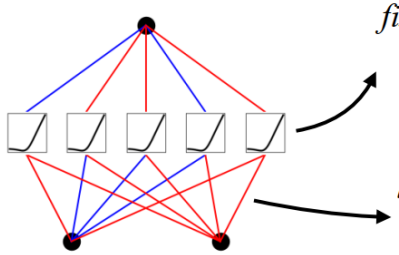
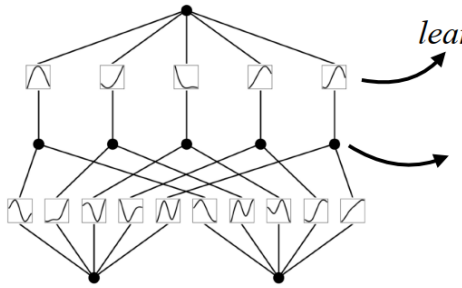
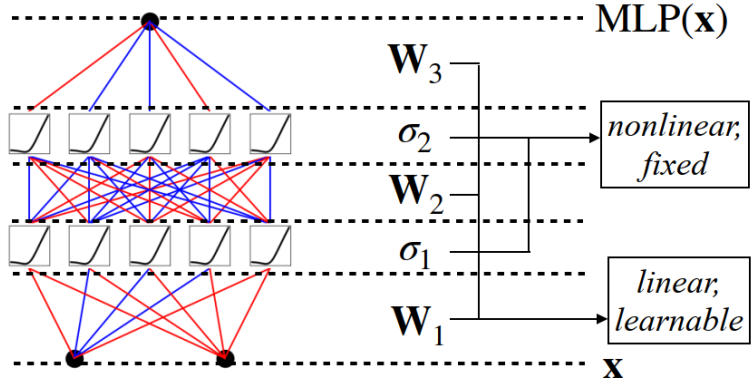
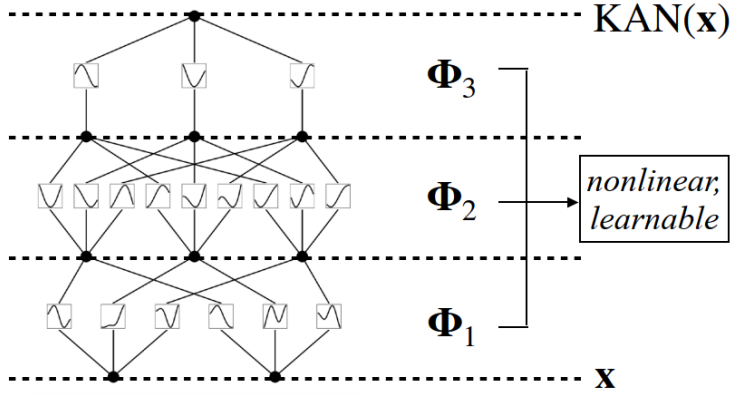
LeCunn: "LLMs are dead end"

Jarek Duda, codes, recording, poster

MLP, KAN – trivial one-way parametrizations … could BNN/we use more?

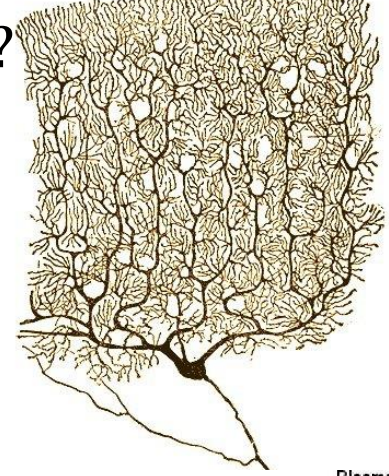
Trained by **backpropagation** – possible for all continuous parametrizations but it is non-local (long connections), biology needs biologically plausible

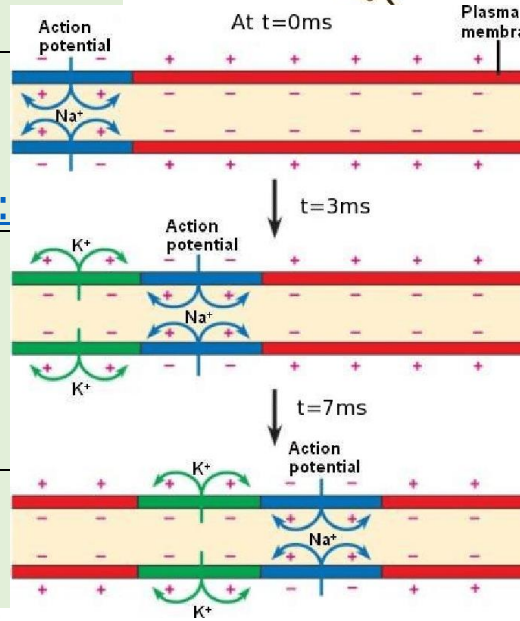
What biology does??? We know only simple e.g. long-term potentiation, Hebbian: “Neurons that fire together, wire together” … **high level mechanisms**???

Model	**Multi-Layer Perceptron (MLP)**	**Kolmogorov-Arnold Network (KAN)**
Theorem	**Universal Approximation Theorem**	**Kolmogorov-Arnold Representation Theorem**
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)	(a) 	(b) 
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)	(c) 	(d) 

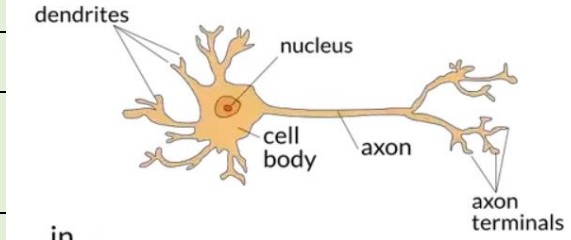
Spiking: **physical level**, but what is **logical**: hidden mathematics?

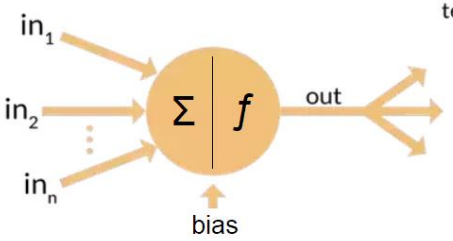
Parameters	**ANN** source	**BNN**
Structure	input weight output hidden layer	dendrites synapse axon cell body (~86B)
Learning	**very precise structures and formatted data**	**can tolerate ambiguity flexible**
Processor	complex high speed one or a few	simple low speed large number
Memory	**separate from a processor** localized non-content addressable	**integrated into processor** distributed content-addressable
Computing	centralized sequential **stored programs**	distributed parallel **self-learning**
Reliability	**very vulnerable**	**robust**
Expertise	numerical and symbolic manipulations	perceptual problems
Operating Environment	**well-defined well-constrained**	**poorly defined un-constrained**





(neurons > RL)





BNN learning ≠ ANN backpropagation

LeCunn: "LLMs are dead end"

Qualitative superiority: just "take more neurons" insufficient?

Spiking: **physical level**, but what is '**logical neuron**': hidden mathematics?

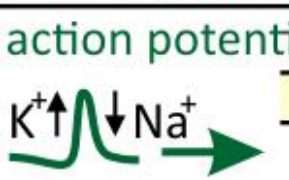
"Rats were found to prefer situations involving certainty to (...) uncertainty"

MLP, KAN parametrizations ← reducible **HCRNN** data structure, **biologically plausible**

optimized for **single direction propagation**

fundamentally **multidirectional (joint distribution)**

action potential



propagate **values** e.g. 3 dogs

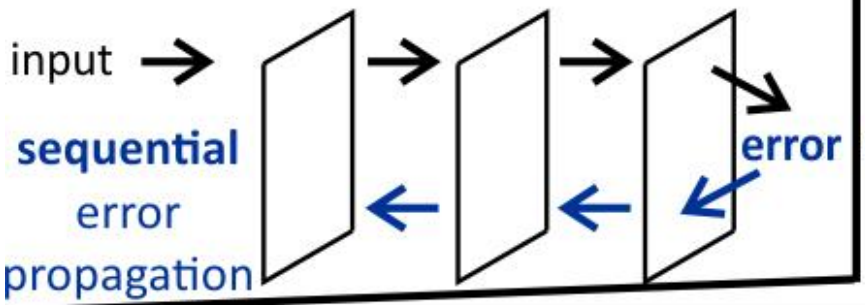
or

propagate **values**

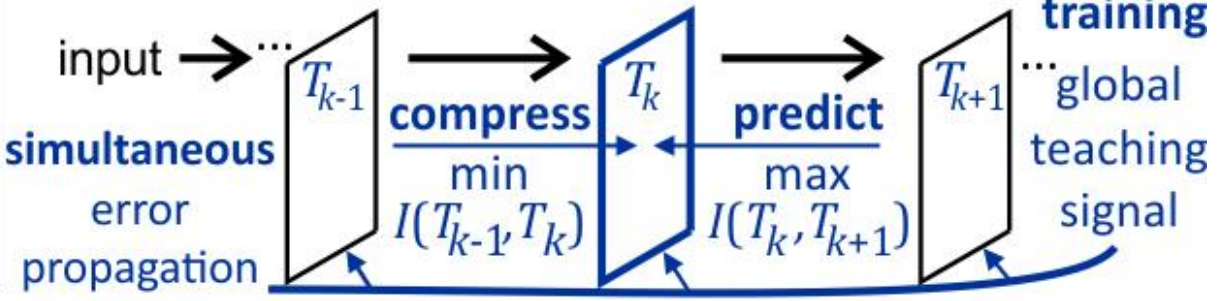
probability distributions (also joint)

e.g. for **uncertainty** estimation

top-down error signals, e.g. backprop



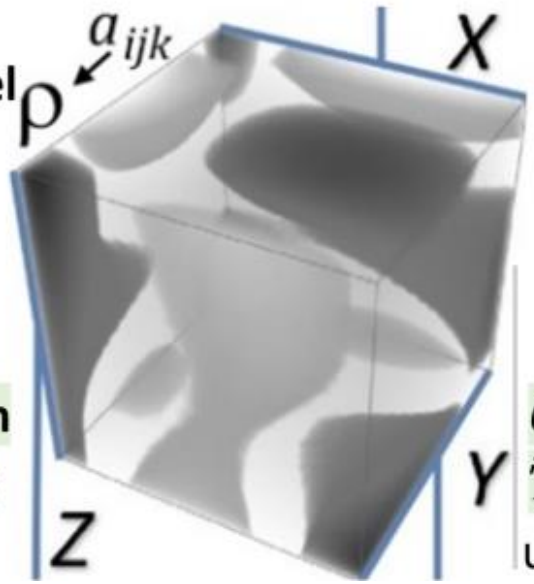
layer-wise error signals, e.g. information bottleneck



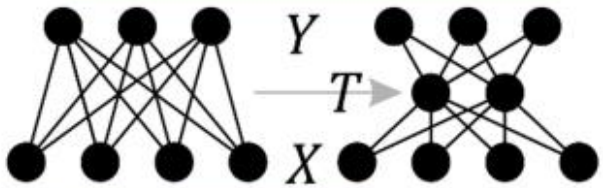
HCRNN – **neurons containing local joint probability distribution model** as $\rho(x) = \sum a_j f_j(x)$ **density**

(a_j) – tensor, $f_j(x)$ – fixed basis

Can be **degenerated to KAN-like** allows **propagation in any direction** of **values**, **probability distributions** e.g. $\rho(x|y,z)$, $\rho(y|x)$, $E[z|x,y]$



trained by backprop, estimation, or **information bottleneck**:



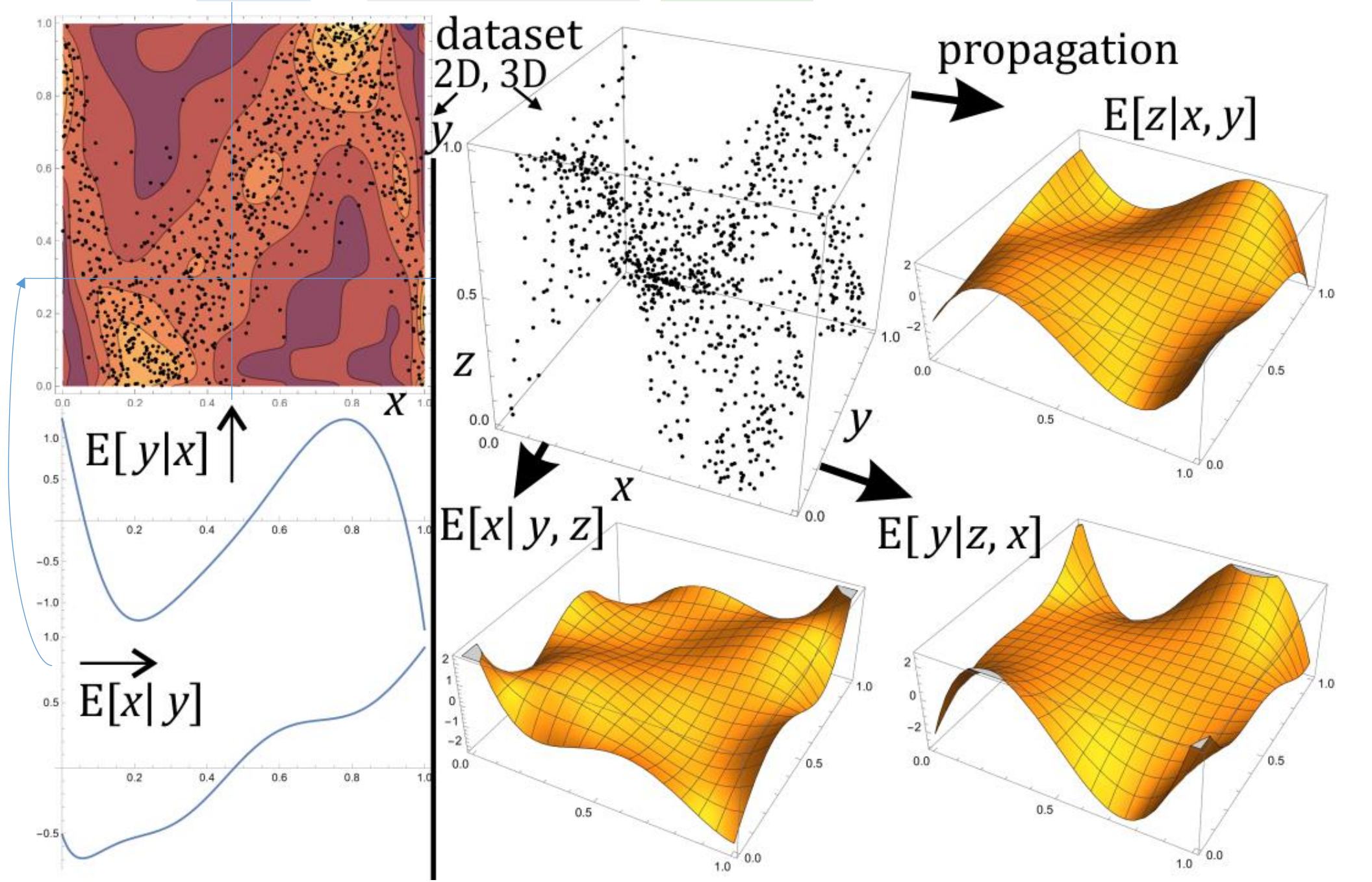
$C_X = \bar{X}\bar{X}^T$, $C_Y = \bar{Y}\bar{Y}^T$ batch features

$C_X - \beta C_Y = O\ \mathrm{diag}(\lambda_1 \le \cdots \le \lambda_n)\ O^T$

$\bar{T} = O_{n\times 1..k}$: k features on size n batch

update NN weights toward: $\bar{X}^T\bar{T}$, $\bar{T}^T\bar{Y}$

Joint distribution neurons as logical for BNNs: multidirectional propagation, of values or probability distribution as vector: ~(expected value, variance, skewness, kurtosis, …)



$d = 2$ variable (a_j) 2W neuron propagating exp. values $(i = 1)$, densities $(i \geq 1)$

Neuron containing local joint distribution model – of its connections:

(a_j) e.g. a_{ij} **matrix** $(d = 2)$ of mixed moments **as neuron parameters**

~KAN: value propagation as polynomial $E[x|y]$ → **2W** $E[y|x]$, **densities**

$$\rho(x,y) = \sum_{i,j\geq 0} a_{ij}\, f_i(x) f_j(y)$$

(a_j) neuron parameters

$$a_{ij} = \frac{1}{|\bar{X}|} \sum_{(x,y)\in\bar{X}} f_i(x)\, f_j(y)$$ direct estimation

$$\rho(x|y) = \sum_{i\geq 0} f_i(x) \frac{\sum_j a_{ij}\ f_j(y)}{\sum_j a_{0j}\ f_j(y)}$$ ~KAN fit poly

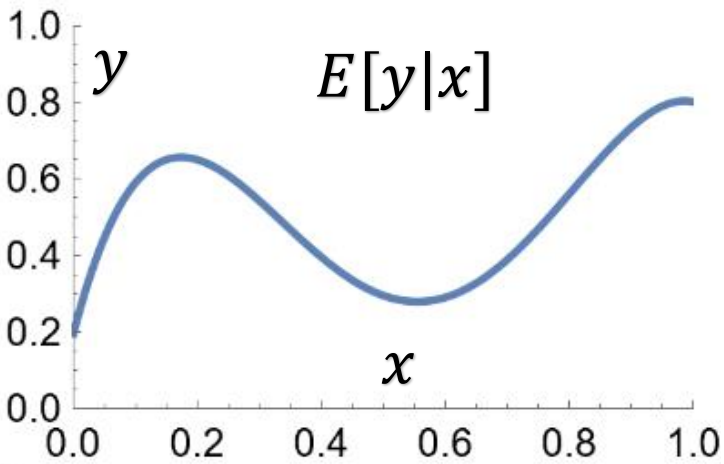
$$E[x|y] = \frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_j a_{1j}\ f_j(y)}{\sum_j a_{0j}\ f_j(y)}$$

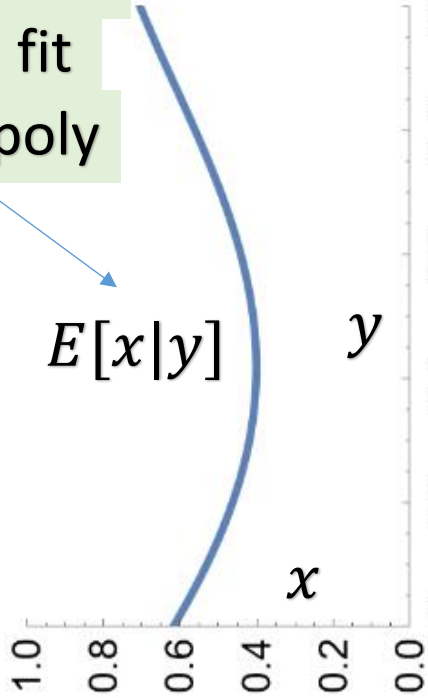
$$\rho(y|x) = \sum_{j\geq 0} f_j(x) \frac{\sum_i a_{ij}\ f_i(x)}{\sum_i a_{i0}\ f_i(x)}$$

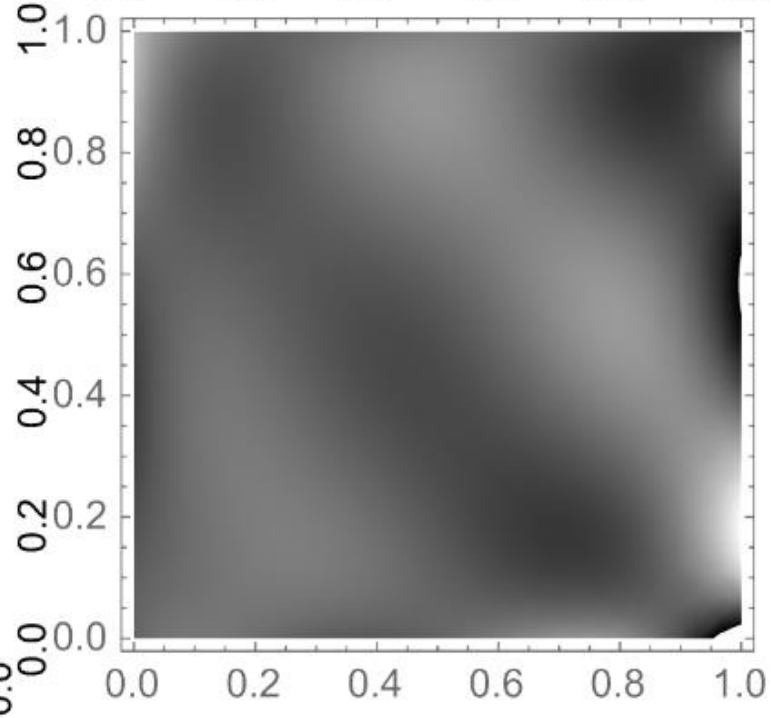
$$E[y|x] = \frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_i a_{i1}\ f_i(x)}{\sum_i a_{i0}\ f_i(x)}$$

$$I(X;Y) \approx \sum_{i.i>0} (a_{ij})^2$$

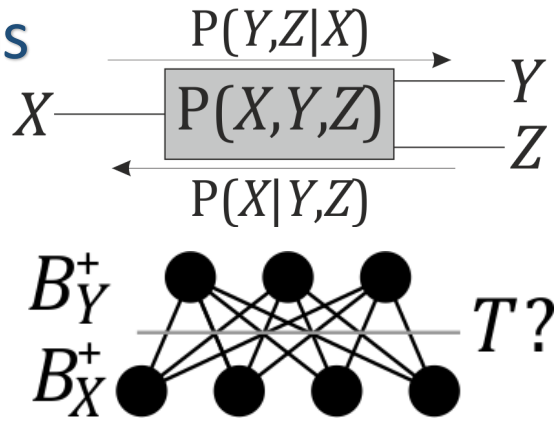
mutual information estimation







Was too abstract – now as KAN response/expansions



Biology-inspired artificial neuron? (2018):

Directly learns conditional probability distributions

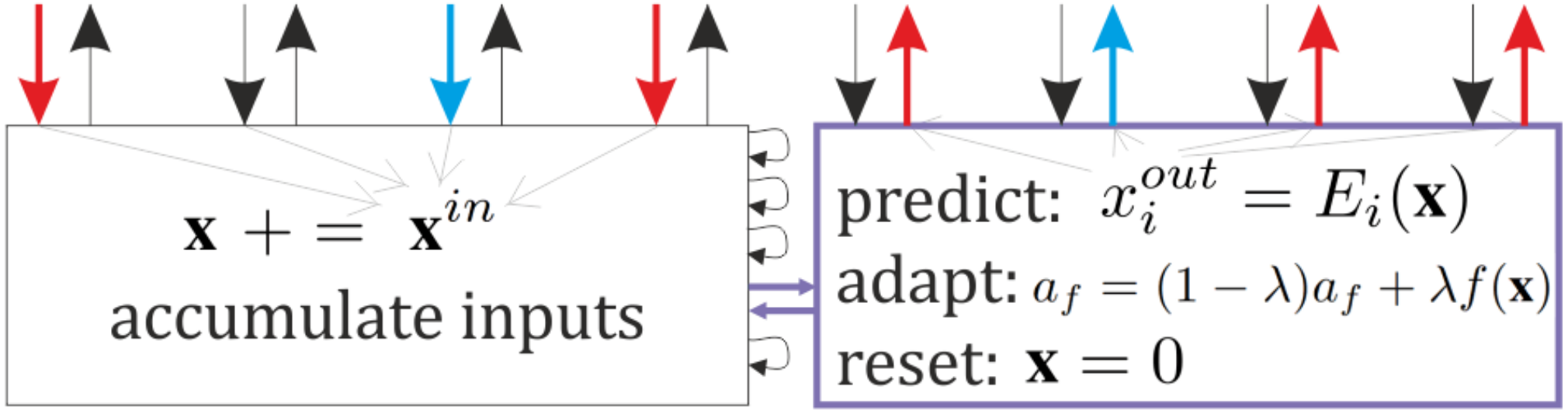
can exploit them by predicting in flexible directions

How to do **deep learning** with them – learn **long correlation chains**?

… how to **train intermediate layers** of hidden dominant features?

Biological neuron: accumulate inputs until trigger, then **impulse**: sends modelled predictions of inputs.

inputs excitatory/inhibitory responses



Can we directly train intermediate layers? (video)

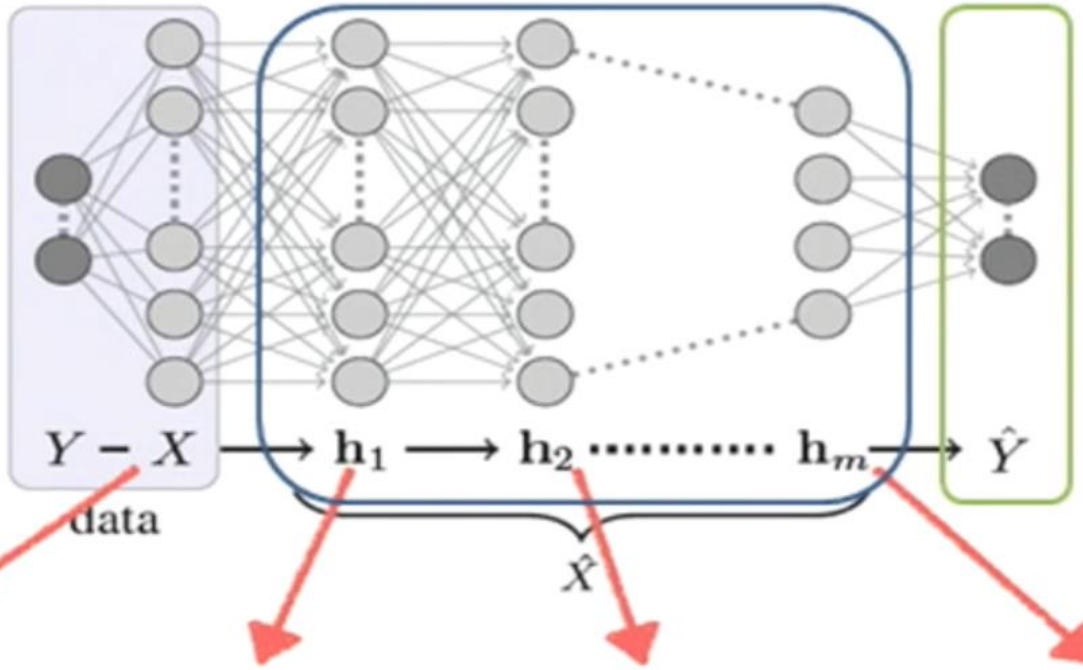
Naftali Tishby, **information theoretic view** – permutation, bijection independent

Markov process between layers, **first extract/compress essential information** reducing **mutual information** [**bits**] $H(X) \geq I(X;T_1) \geq I(X;T_2) \geq \cdots$

Information bottleneck (Tishby): for $X \to T \to Y$ optimize $\inf_T I(X;T) - \beta I(T;Y)$

$\inf_T I(X;T)$

compression



$\sup_T I(T;Y)$

prediction

$I \sim \mathrm{HSIC} = \mathrm{Tr}(K_x, K_y)$

$K_x(x,x') = \langle \phi(x), \phi(x') \rangle$

$n \times n$ as **sample size**

ϕ Gaussian: **local, width?**

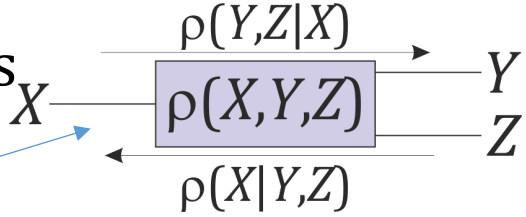
HCR: **many global** poly (f_j) **for normalized**

decompose dependencies into mixed moments – control, accuracy, credibilty

HIERARCHICAL CORRELATION RECONSTRUCTION

for time series, conditional distribution (~~Bayes~~) models

(nonlinear, adaptive, all-directional) artificial neurons



How to model/estimate density from a data sample?

MSE fit polynomial $\rho(x) = \sum_{f\in B} a_f f(x)$ (in (f) orthonormal basis)

also for **joint distribution**, **non-stationarity**, **missing data**

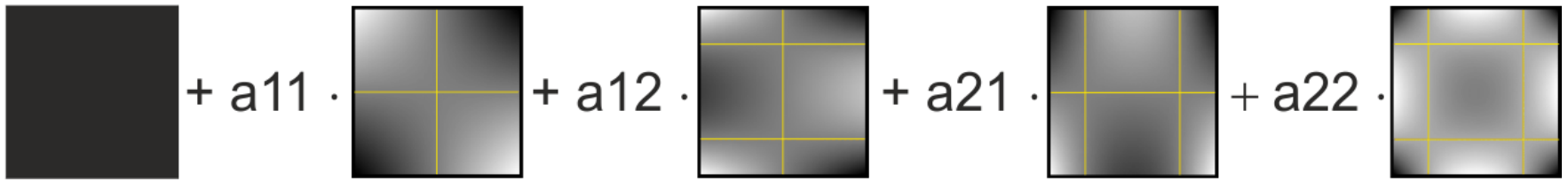
	Moments/cumulants	$\boldsymbol{\rho(x) = \sum_f a_f f(x)}$	**Machine learning, NN**
# parameters	low – rough	from low to high	high - accurate
estimation	e.g. $m_k = \frac{1}{\lvert X\rvert}\sum_{x\in X} x^k$	$a_f = \frac{1}{\lvert X\rvert}\sum_{x\in X} f(x)$	usually iteration
Interpretable?	yes	Yes: mixed moments	depends, **trust**???
Independently?	yes	Yes (adapt, missing)	depends
Unique?	yes	yes (MSE)	often huge freedom
Accuracy?	controllable	controllable	usually uncontrollable
Density?	moment problem	**YES:** $\sum_f a_f f(x)$	depends
→ complete	depends	yes	depends

(intro, talk, slides)

~homoscedasticity

each variable normalized to ~uniform

pair-wise joint density ≈ independent + a11 · ~ correlation coef. + a12 · + a21 · + a22 ·



extract, exploit additional information from data, easy to publish

Some articles using **hierarchical correlation reconstruction**: introduction with Mathematica code

[1] J. Duda, Rapid parametric **density estimation**, arXiv:1702.02144 (2017)

[2] J. Duda, **Hierarchical correlation reconstruction** with missing data, for example for **biology-inspired neuron**, arXiv:1804.06218 (2018)

[3] J. Duda, **Exploiting statistical dependencies of time series** with hierarchical correlation reconstruction, arXiv:1807.04119 (2018)

[4] J. Duda, M. Snarska, Modeling joint probability distribution of **yield curve parameters**, arXiv:1807.11743

[5] J. Duda, A. Szulc, **Credibility evaluation** of income data with hierarchical correlation reconstruction, arXiv:1812.08040 (2018), International Conference on Applied Economics (2020)

[6] J. Duda, R. Syrek, H. Gurgul, Modelling **bid-ask spread conditional distributions** using hierarchical correlation reconstruction, arXiv:1911.02361(2019), Statistics in Transition vol 21 no 4 (2020)

[7] J. Duda, G. Bhatta, Log-stable probability density functions, **non-stationarity evaluation**, and **multi-feature autocorrelation analysis** of the γ-ray light curves of blazars, arXiv:2005.14040 (2020), Monthly Notices of the Royal Astronomical Society Main Journal (2021)

[8] J. Duda, H Gurgul, R. Syrek, Multi-feature evaluation of **financial contagion**, CEJOR (2021)

[9] J. Duda, Predicting **conditional probability distributions** of redshifts of Active Galactic Nuclei using Hierarchical Correlation Reconstruction, arXiv:2206.06194 (2022), Monthly Notices of the Royal Astronomical Society Main Journal (2024)

[10] J. Duda, S. Podlewska, Low cost **prediction of probability distributions** of molecular properties for early virtual screening, arXiv:2207.11174, Molecular Diversity (2022)

[11] J. Duda, **Time delay multi-feature correlation analysis** to extract subtle dependencies from **EEG signals**, arXiv:2305.09478 (2023)

[12] J. Duda, **Extracting individual variable information** for their decoupling, direct mutual information and multi-feature Granger causality, *arXiv:2311.13431* (2023)

[13] J. Duda, J. Leśkow, P. Pawlik, W. Cioch, CMAFI — Copula-based Multifeature Autocorrelation Fault **Identification of rolling bearing**, Mechanical Systems and Signal Processing (2024)

[14] J. Duda, Biology-inspired **joint distribution neurons** based on Hierarchical Correlation Reconstruction allowing for **multidirectional neural networks**, *arXiv:2405.05097* (2024)

[15] J. Strawa, J. Duda, **Improving KAN with CDF normalization to quantiles**, *arXiv:2405.13393* (2025)

[16] J. Duda, J. Bracha, A. Przybysz, … **independence test** of similar performance as **HSIC** , *arXiv:2508.18338*

sociology

finance

astronomy

chemoinformatics

EEG

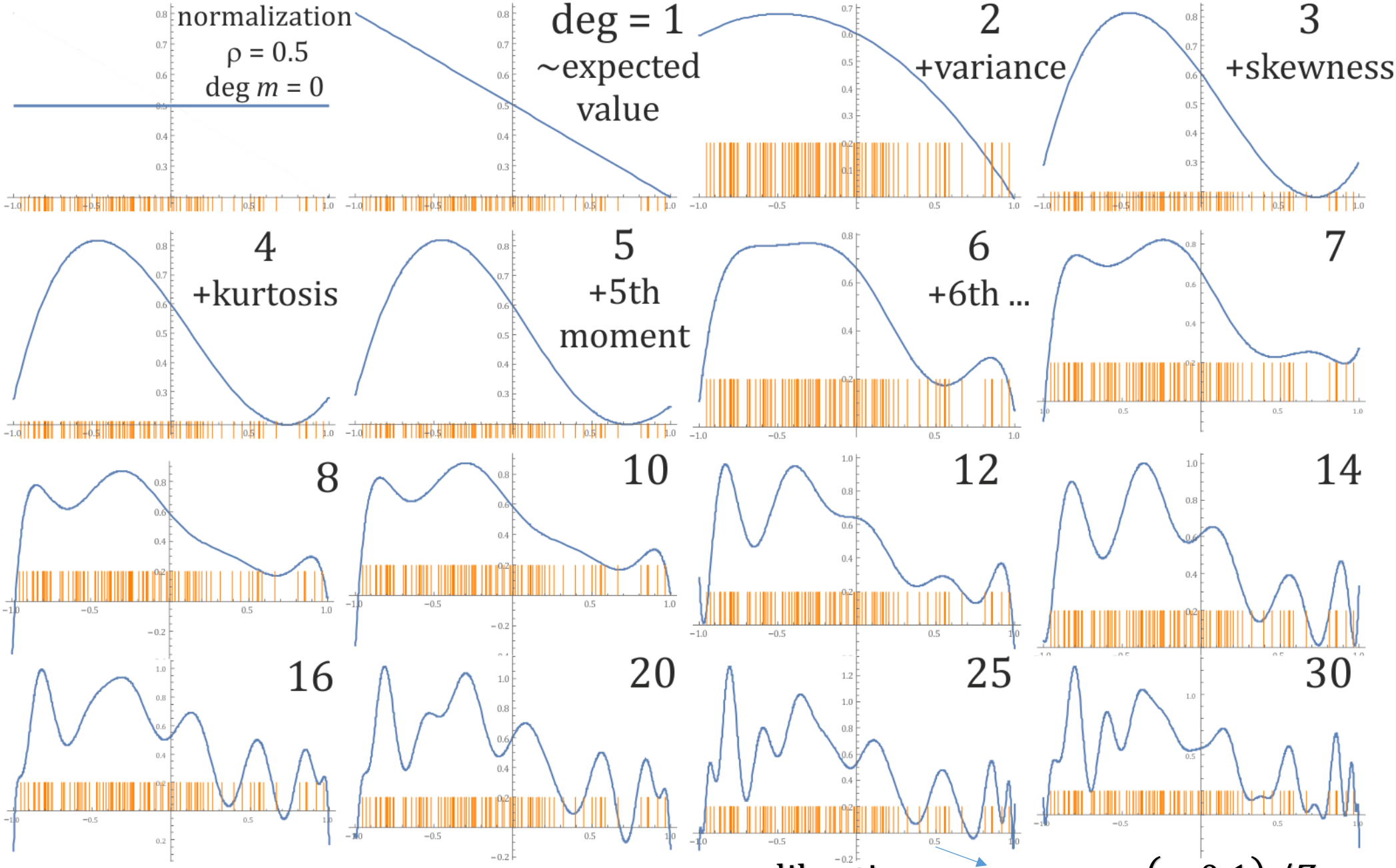
mechanics

Most “single neuron” – let’s build networks from them

Issue: density as polynomial can get below 0, seems negligible for NNs?

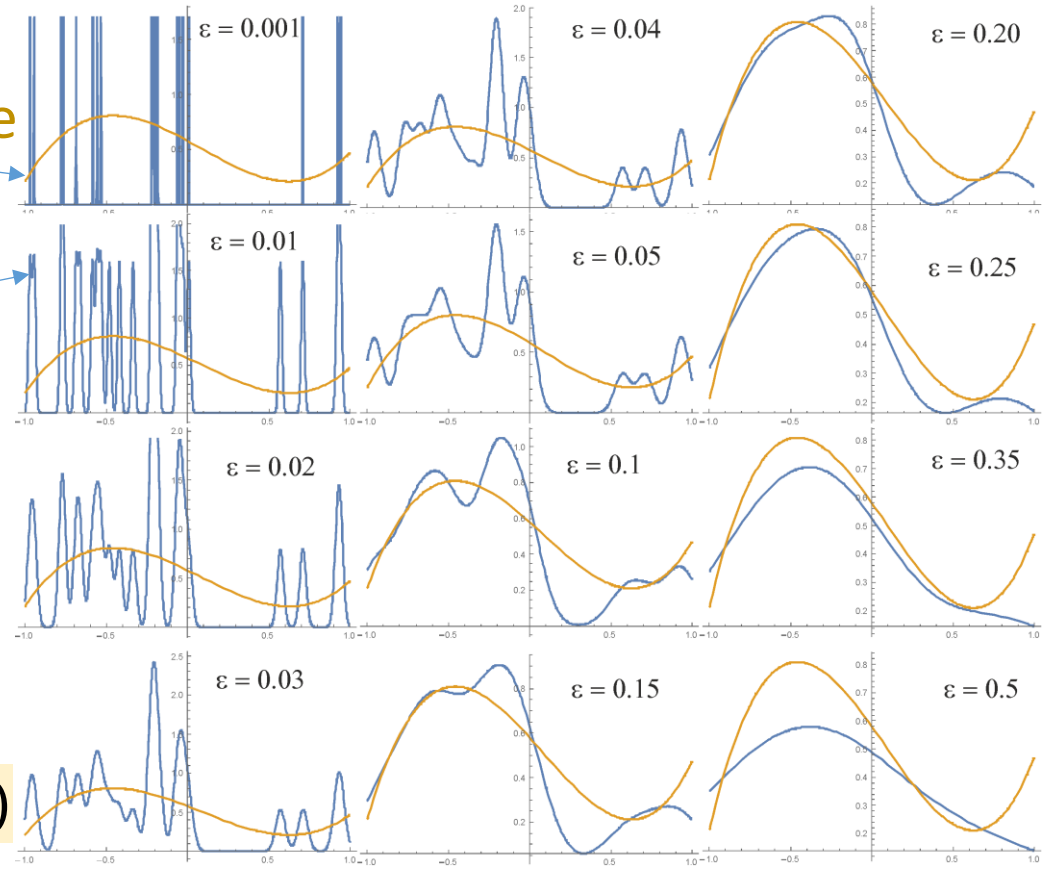
$n = 100$ size **1D sample** (from degree = 3), **density estimated as polynomial**:

on [-1,1] $\approx$ (deg $m \to \infty$ leads to sum of Dirac deltas)



calibration e. g. $\rho \to \max(\rho, 0.1)/Z$

Derivation: $n = 25$ size sample



KDE (kernel density estimation):

g_ϵ : ϵ-width Gaussian in each point

Find $\rho_{\boldsymbol{a}}(\boldsymbol{x}) = \sum_{\boldsymbol{j}} a_{\boldsymbol{j}} f_{\boldsymbol{j}}(x)$ minim. MSE

$$\arg\min_{\boldsymbol{a}} \int (\rho_{\boldsymbol{a}} - g_\epsilon)^2 dx =$$

$$\arg\min_{\boldsymbol{a}} \|\rho_{\boldsymbol{a}}\|^2 - 2\langle \rho_{\boldsymbol{a}}, g_\epsilon \rangle + \|g_\epsilon\|^2$$

Taking $\epsilon \to 0$, $\langle \rho_{\boldsymbol{a}}, g_\epsilon \rangle = \sum_{x \in X} \rho_{\boldsymbol{a}}(x)$

Removing $\lim_{\epsilon \to 0} \|g_\epsilon\|^2 = \infty$ which does not affect parameters $\boldsymbol{a}$

Using **orthonormal**: $\langle f_i, f_j \rangle = \int f_i(x) f_j(x) dx = \delta_{ij}$ e.g. on $[0,1]^d$

$$\arg\min_{\boldsymbol{a}} \|\rho_{\boldsymbol{a}}\|^2 - \frac{2}{n} \sum_{x \in X} \rho_{\boldsymbol{a}}(x) = \arg\min_{\boldsymbol{a}} \sum_{\boldsymbol{j}} (a_{\boldsymbol{j}})^2 - \frac{2}{n} \sum_{x \in X} \sum_{\boldsymbol{j} \in B} a_{\boldsymbol{j}} f_{\boldsymbol{j}}(x)$$

$$\text{minimum: } \partial_{a_j} = 0 \quad \Rightarrow \quad a_{\boldsymbol{j}} = \frac{1}{n} \sum_{x \in X} f_{\boldsymbol{j}}(x)$$

Normalize for polynomial approximation

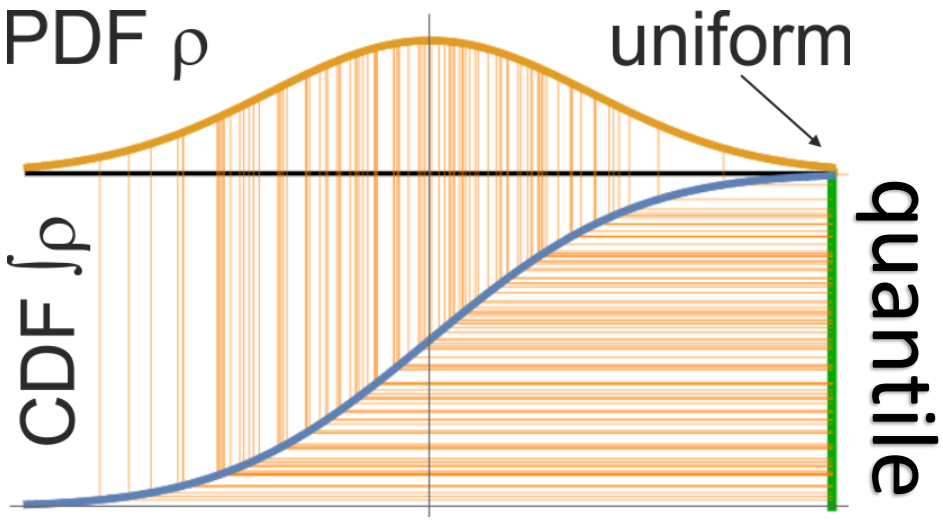
In practice: **normalize each variable**

to ~ **uniform distribution:** $x^t = \mathbf{CDF}(y^t)$

(1/2: median, position: quantile, like **copula**)

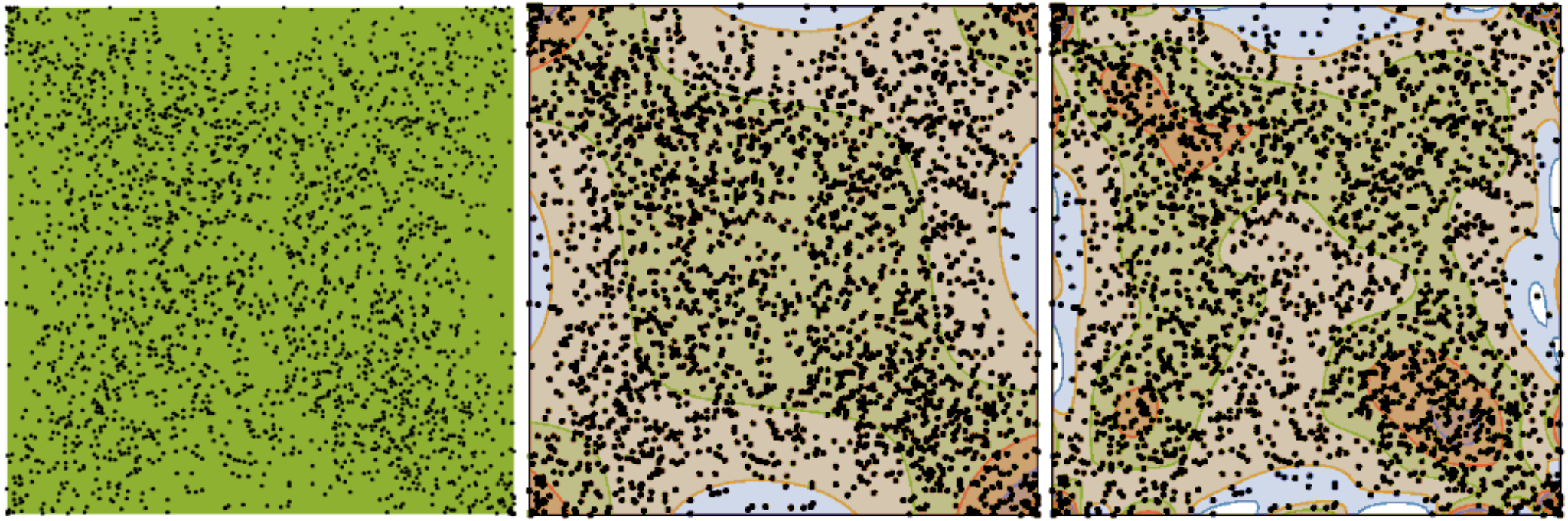
Then fit polynomial as joint distribution

(daily log returns: $\ln(v_{t+1}/v_t)$)

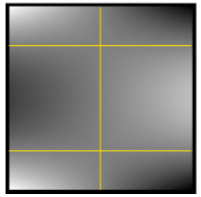
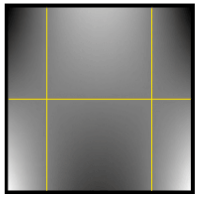
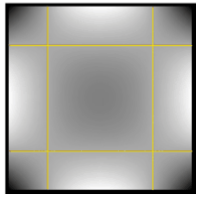


(x^{t-1},x^t) pairs from TRV series + used estimated density ρ isolines

MLE κ, m=0, ρ=1 | HCR m=2 | 0, 0.5, 1, 1.5, 2, 2.5 | HCR m=9



independent | ~ correlation coef. | **further statistical dependencies** | var-var

pair-wise joint density $\approx$  $+ a_{11} \cdot$  $+ a_{12} \cdot$  $+ a_{21} \cdot$  $+ a_{22} \cdot$ 

Gaussian · Hermite polynomials alternative normalization – e.g. for shape descriptors

detailed continuous decodable rotation-invariant shape descriptors

shape features of e.g. molecules, **shape similarity metric** avoiding rotation optimization

arXiv:2601.03326

PCA can approximate **shape** as **ellipsoid**

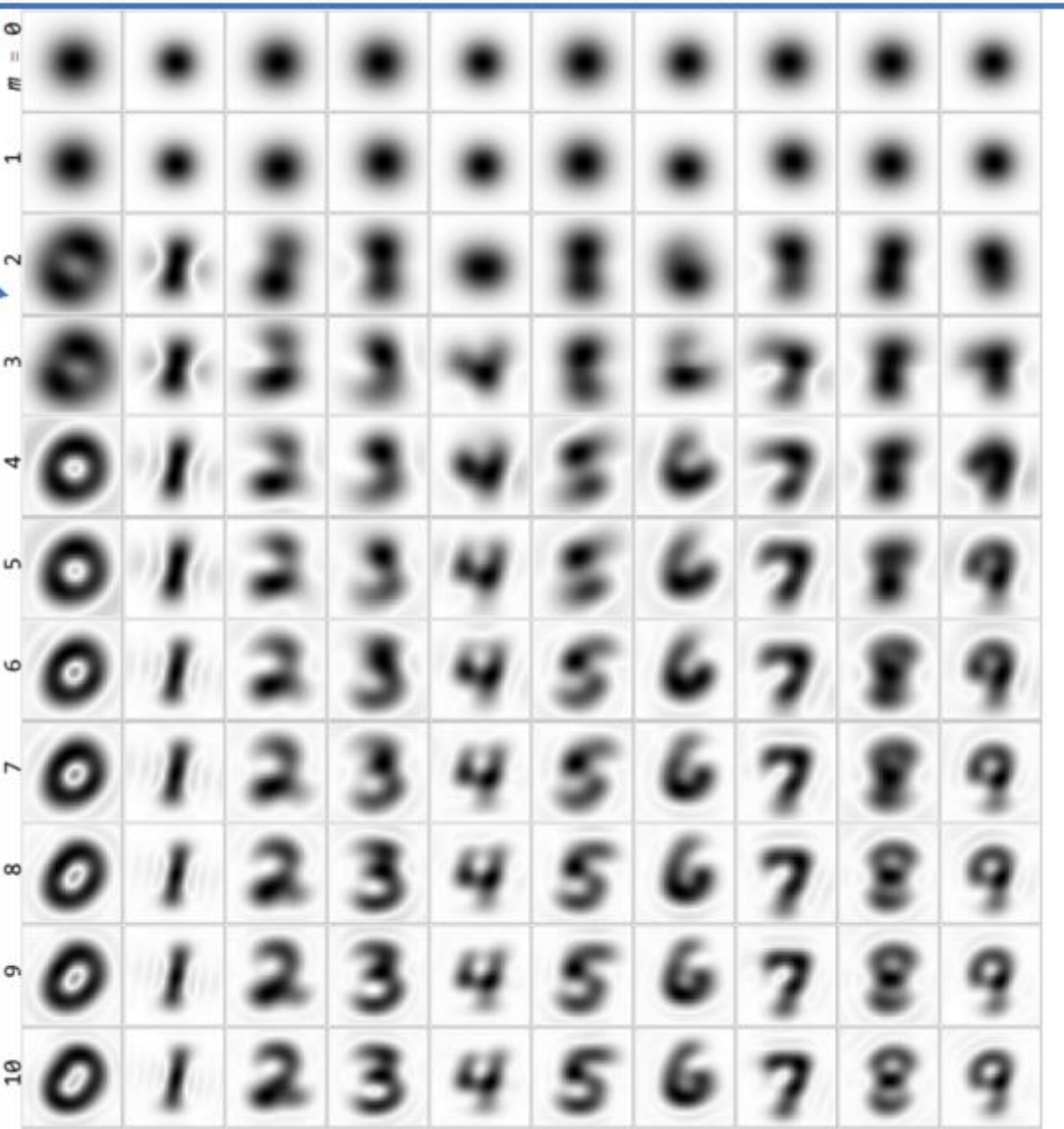
$[p]_{ab} = \mathrm{E}[(x_a - \mathrm{E}[x_a])(x_b - \mathrm{E}[x_b])]$

$\mathrm{Tr}([p]^{m=1,\dots,d})$ are **rotation invariants**

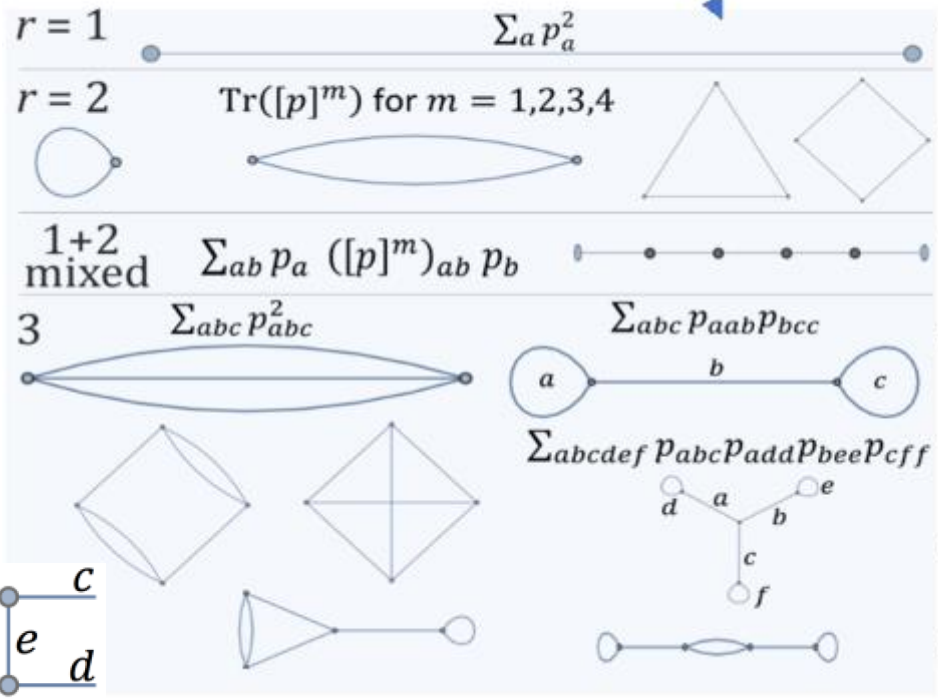
Extension: **detailed tensor description**:

$\mathrm{E}[(x_a - \mathrm{E}[x_a])(x_b - \mathrm{E}[x_b])(x_c - \mathrm{E}[x_c])]$

or **decodable**: **Gaussian · polynomial**



Detailed continuous rotation invariants



Generalize? global vs local basis

KDE – kernel density estimation

e.g. ϵ-radius Gaussian in each point

huge #parameters ~ #points

how to choose (ellipse?) radii??

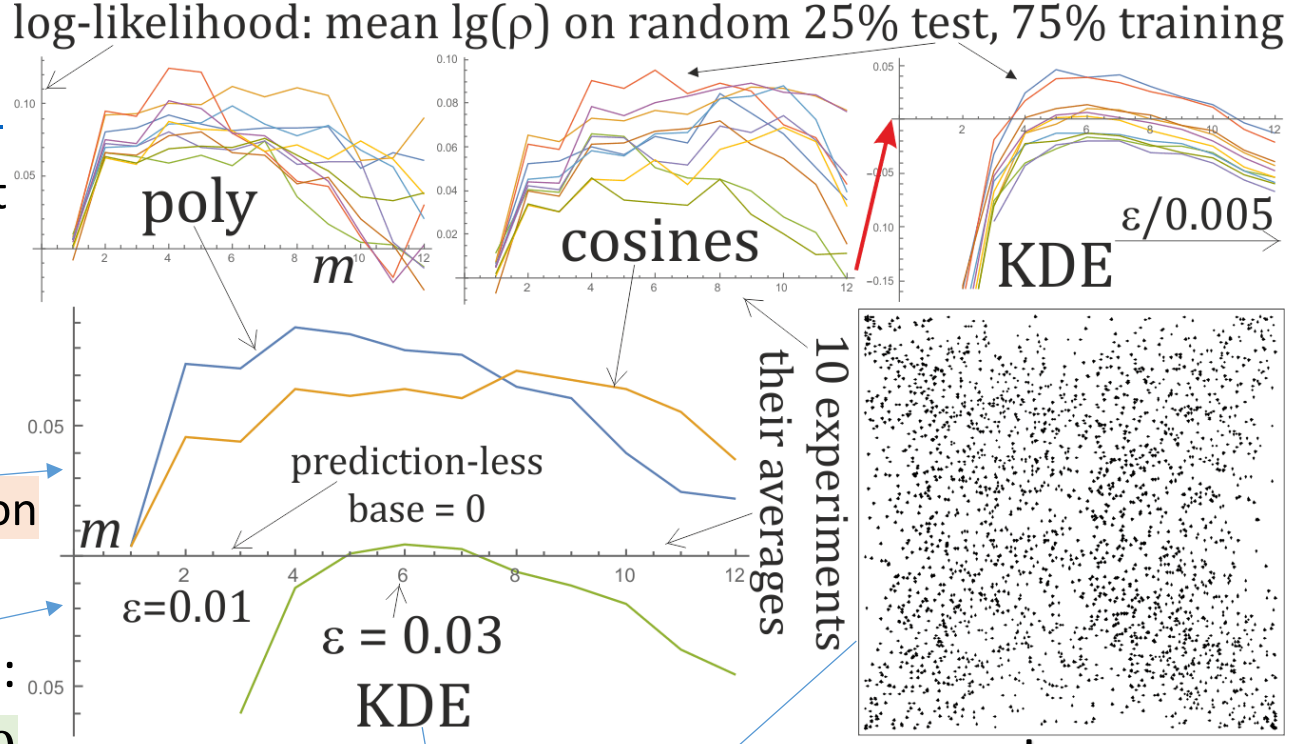
doesn't work in high dimension

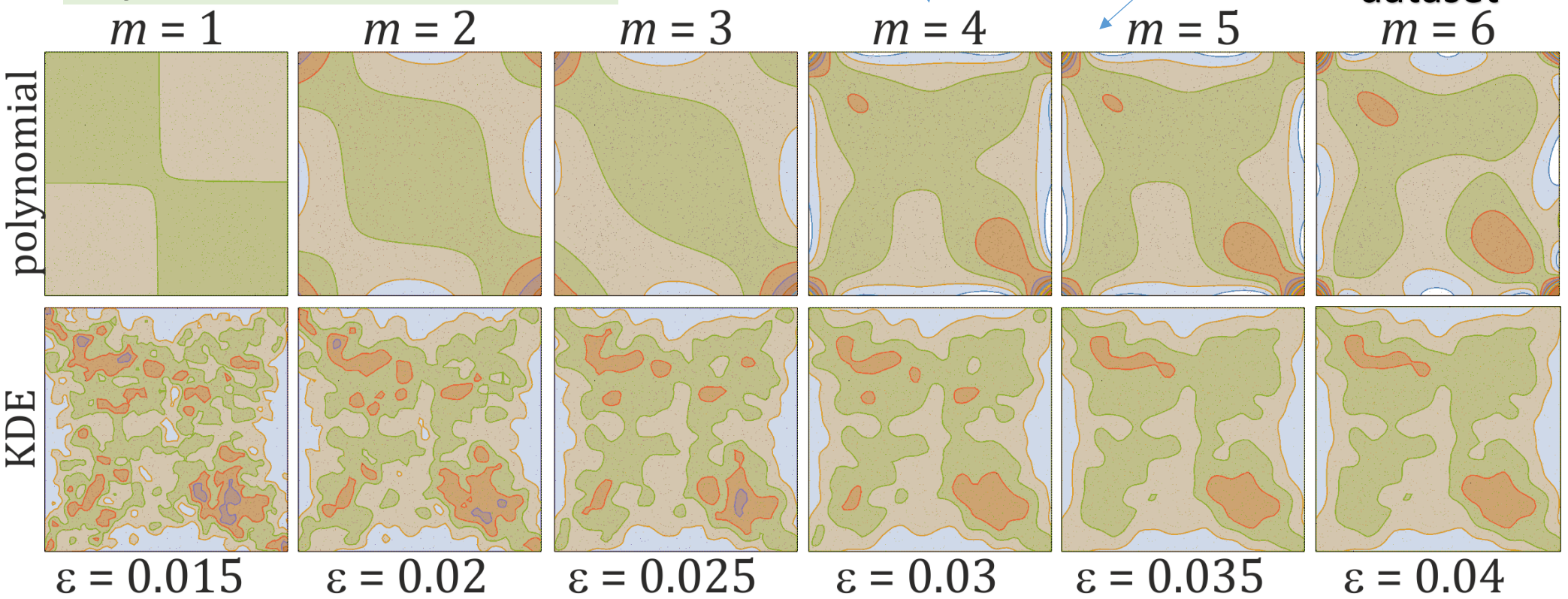
terrible log-likelihood, generalization

as it localizes in the old points

cross-validation:

Polynomial: MSE fitted to $\epsilon \to 0$





Hierarchical correlation reconstruction: $\rho(\boldsymbol{x}) = \Sigma_j\, a_j \boldsymbol{f_j}(\boldsymbol{x})$

d variables, up to m-th moment, number of a_j moments: $(m+1)^d = \sum_{k=0}^{d} \binom{d}{k} m^k = 1 + dm + \frac{1}{2}d(d-1)m^2 + \cdots$ (marginal, pairwise, normalization)

$d = 3$

$d = 1 \rightarrow$

$d = 2 \searrow$

$f_i(x)\, f_j(y)$

$f_0(x) = 1$ normalization

$f_1(x)$ ~exp. value

$f_2(x)$ ~variance

$f_3(x)$ ~skewness

$f_4(x)$ ~kurtosis

$f_0(y) = 1$ normalization

$f_1(y)$

$f_2(y)$

$f_3(y)$ zero

$f_4(y)$

a_{000} normalize

a_{001} marginal

a_{010} marginal

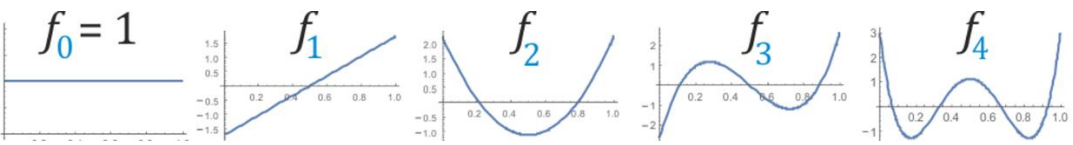
a_{100} marginal

a_{011} pairwise

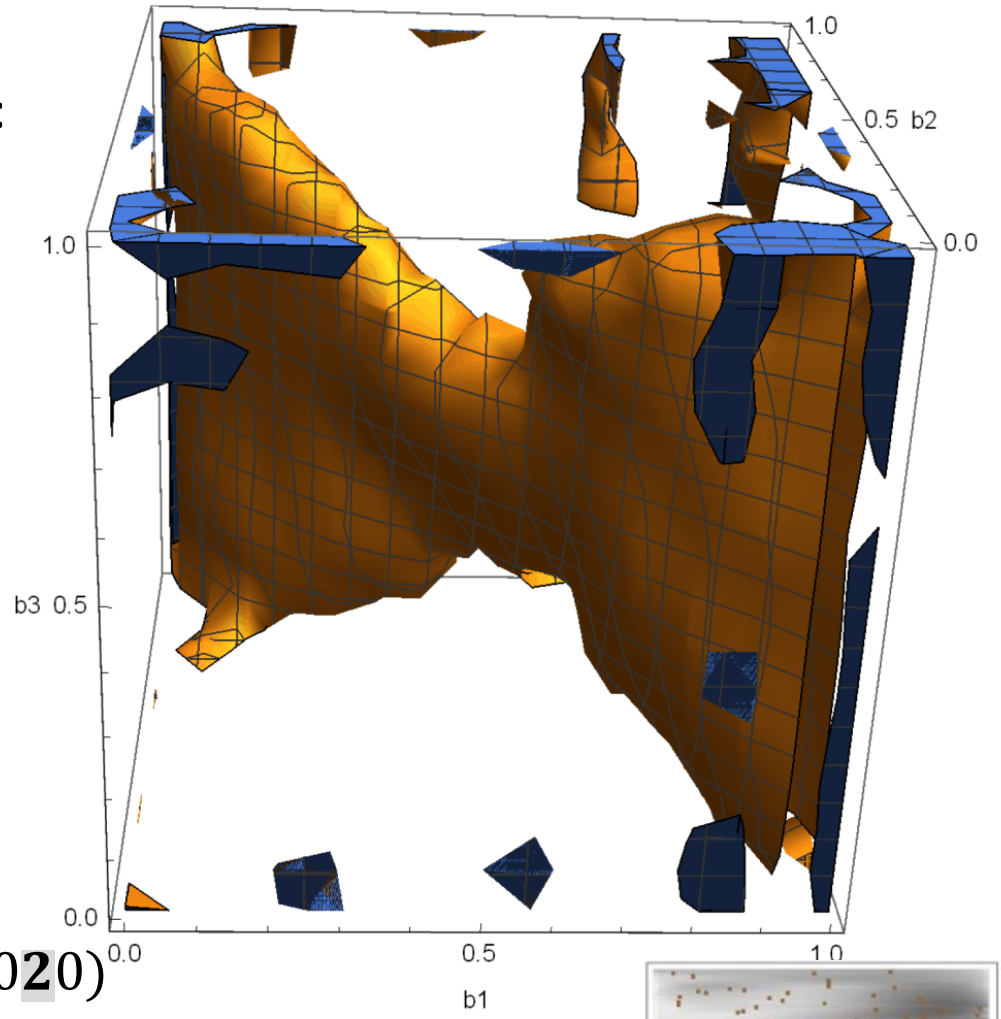
a_{101} pairwise

a_{110} pairwise

a_{111} triplewise



ρ > 2 region: ~14% of volume, ~62% of cases:



Also $[0,1]^d$ in **higher dimensions**, e.g. $d = 3$:

$$\rho(x_1, x_2, x_3) = \sum_{j \in B} a_j \; f_{j_1}(x_1) f_{j_2}(x_2) f_{j_3}(x_3)$$

⇒ **conditional distributions without Bayes**

MSE estimated from dataset $X \subset \mathbb{R}^3$:

$$a_j = \frac{1}{|X|} \sum_{x \in X} f_{j_1}(x_1) f_{j_2}(x_2) f_{j_3}(x_3)$$

For considered **statistical dependencies:**

basis B of **considered mixed moments**

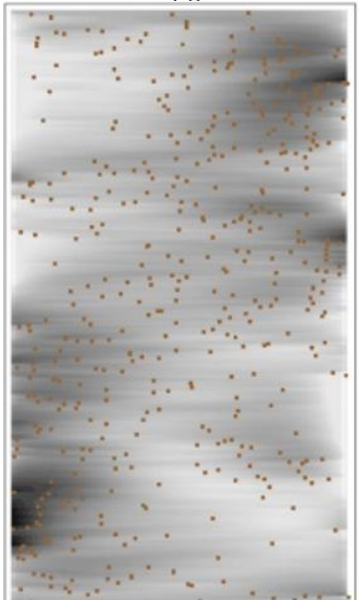
a_j describes e.g. **variance-variance between**

hierarchical e.g. $d = 9$: $B \ni \boldsymbol{j} = (000\mathbf{2}000\mathbf{2}0)$

a_j = **average** of $f_{j_1}(x_1) \ldots f_{j_d}(x_d)$ over dataset - entire or:

- over a subset for **missing data**: need only $j > 0$ coord. as $f_0 = 1$
- $a_j^{t+1} = \eta a_j + (1-\eta) f_j(\boldsymbol{x})$ parameter evolution in **nonstationary**

  e.g. $a_j(t+1) = a_j(t) + (1-\eta)\left(f_j(x_t) - a_j(t)\right)$ for $j = 1,2,3,4$



Uniform projected to horizontal, model evolution (gray):

Value propagation $E[x|y,\dots]$ from HCR joint density $\rho(x,y,\dots)$ e.g. **neuron**:

Especially for **expected values**, let us **ignore the $\rho<0$ issue** – simplifying a lot

$$d=2: \qquad \rho(x,y)=\sum_{ij} a_{ij}\, f_i(x) f_j(y) \qquad \Rightarrow$$

substitute y and normalize

$$\rho(x|y)=\frac{\sum_{ij} a_{ij}\ f_i(x) f_j(y)}{\int_0^1 \sum_{ij} a_{ij}\ f_i(x) f_j(y)\,dy}=\sum_{i=1,..} f_i(x)\,\frac{\sum_j a_{ij}\, f_j(y)}{\sum_j a_{0j}\, f_j(y)}$$

restrict to 1st moment

moments

$$E[x|y]=\frac{1}{2}+\frac{1}{2\sqrt{3}}\,\frac{\sum_j \boldsymbol{a_{1j}}\, \boldsymbol{f_j(y)}}{\sum_j a_{0j}\, f_j(y)} \qquad \text{as} \qquad \int_0^1 x f_0(x)dx=\frac{1}{2},\ \int_0^1 x f_1(x)dx=\frac{1}{2\sqrt{3}}$$

can propagate in any direction e.g.: $E[y|x]=\frac{1}{2}+\frac{1}{2\sqrt{3}}\,\frac{\sum_j \boldsymbol{a_{j1}}\, \boldsymbol{f_j(x)}}{\sum_j a_{j0}\, f_j(x)}$

$$E[x|y,z]=\frac{1}{2}+\frac{1}{2\sqrt{3}}\,\frac{\sum_{jk} a_{1jk}\, f_j(y)\, f_k(z)}{\sum_{jk} a_{0jk}\, f_j(y)\, f_k(z)} \xrightarrow[\text{normal.}]{\text{pairwise}} \sum_j a_{1j0} f_j(y)+a_{10j} f_j(z)$$

~KAN, trained poly.

parameters: moments

Polynomial summation, normalize e.g. $\phi(y)=\mathrm{CDF}\left(a_{1j0} f_j(y)\right)$ for **minibatch**

degenerating to KAN for **pairwise**, allowing to consciously add triplewise, higher

… and **adding many new training possibilities** …

Biological plausibility? Summation + trained 1-parameter functions?

Poly. KAN with CDF norm.:

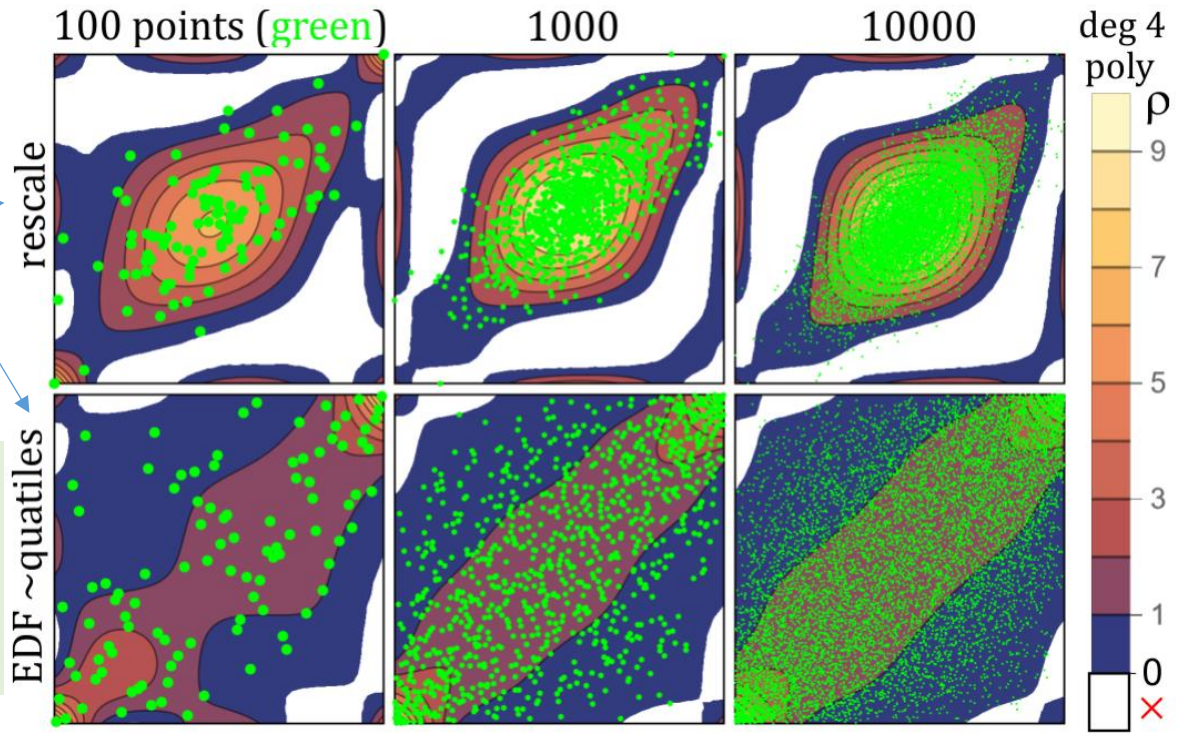
arXiv:2507.13393 –

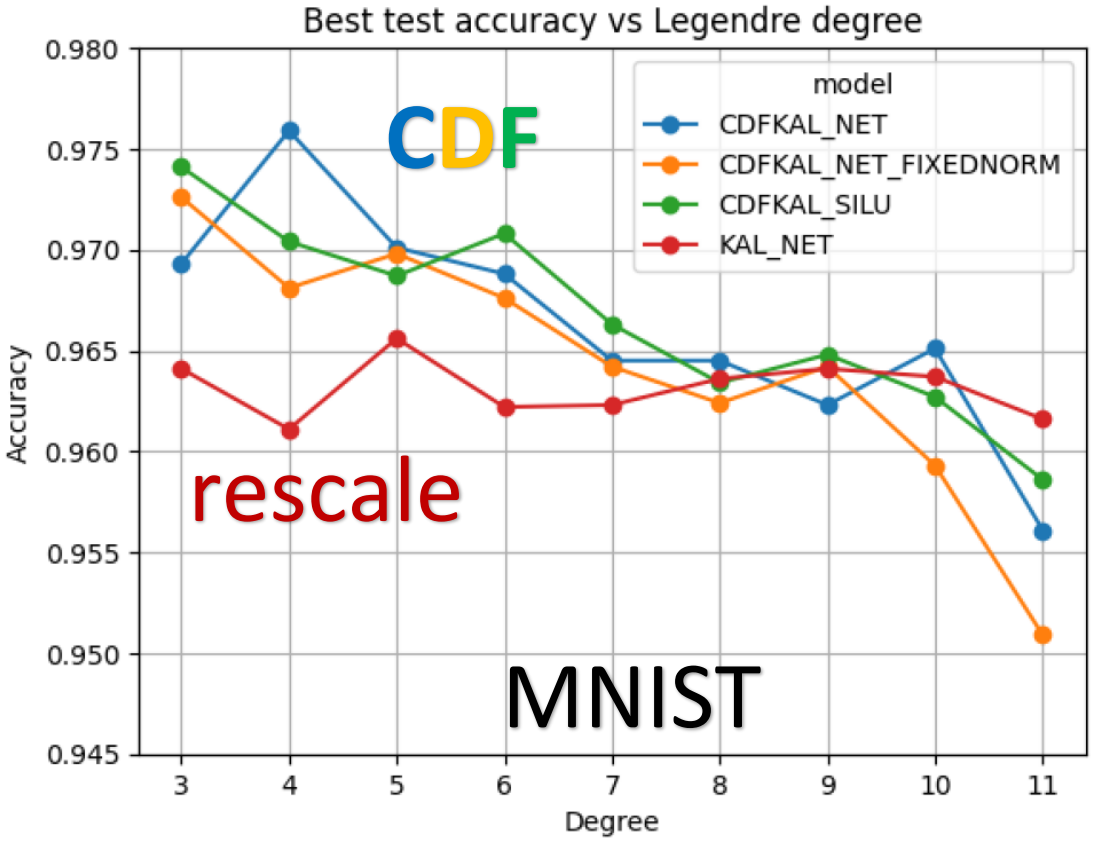
improving Legendre KAN

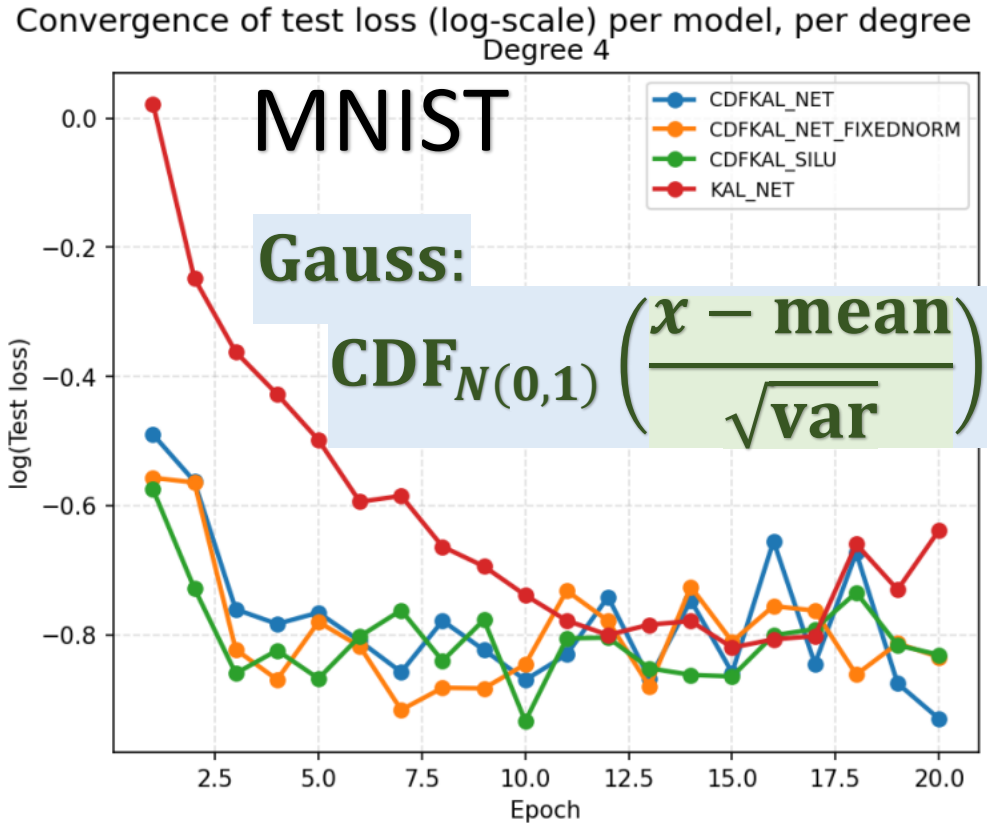
by replacing **rescale** with

CDF/EDF normalization

More appropriate for low degree descriptions – better at generalization





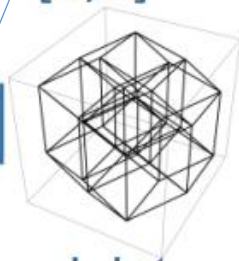


1 neuron poly KAN trained by **direct estimation**: matrix product

```
f[x_] := Exp[x[[1]] - x[[2]]^2 - x[[3]]^3 + x[[4]]^4]; d = 4; (* function, variables *)
m = 4; n = 1000; SeedRandom[1]; (* max. features, points, next find basis: *)
fs = Expand[Table[LegendreP[i, x] * Sqrt[(2 i + 1)] /. x → 2 x - 1, {i, m}]];
X = RandomVariate[NormalDistribution[0, 1], {n, d}]; Y = Map[f, X];
inX = Table[normEDF[X[[All, i]]], {i, d}]; inY = normEDF[Y]; (*normalization*)
nX = Transpose[Table[inX[[i]][X[[All, i]]], {i, d}]]; nY = inY[Y];
md = 4; X̄ = Flatten[fs[[1 ;; md]] /. x → nX, {{2}, {3, 1}}]; (* features *)
mu = 1; Ȳ = Transpose[fs[[1 ;; mu]] /. x → nY]; (* only expected value *)
as = Flatten[Transpose[X̄].Ȳ] / n; (* direct parameter estimation *)
cp = 1 / 2 + Expand[Partition[as, md].fs[[1 ;; md]]] / Sqrt[12]; (* E *)
Print[Row[{ListPlot[Transpose[{nY, Y}]], "of sum of below: "}]];
Row[Table[cf = cp[[i]] /. x → inX[[i]][x]; (* polynomials(normalized X) *)
  Plot[cf, {x, -2, 2}, ImageSize → 130], {i, d}]]
```

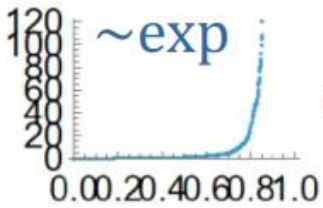
4+1

[0,1]



joint distr. model

trained

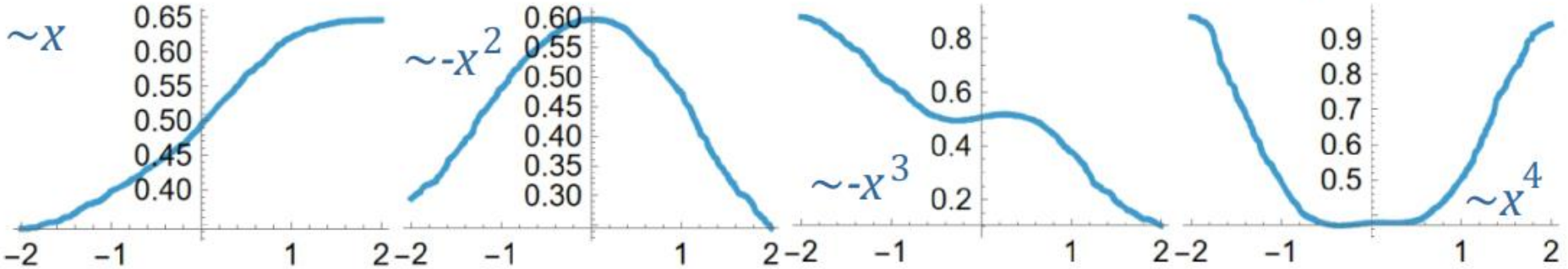


of sum of below:

KAN-like example: 1-paremeter function, summation

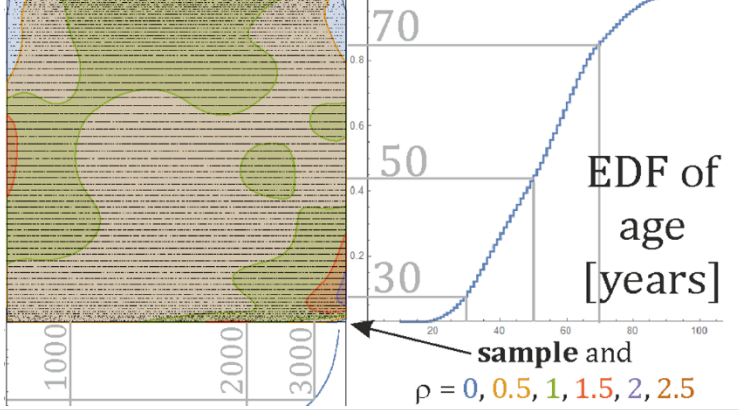
$E[x|y,z] \approx \text{normalized } \sum_j a_{1j0} f_j(y) + \sum_j a_{10j} f_j(z)$

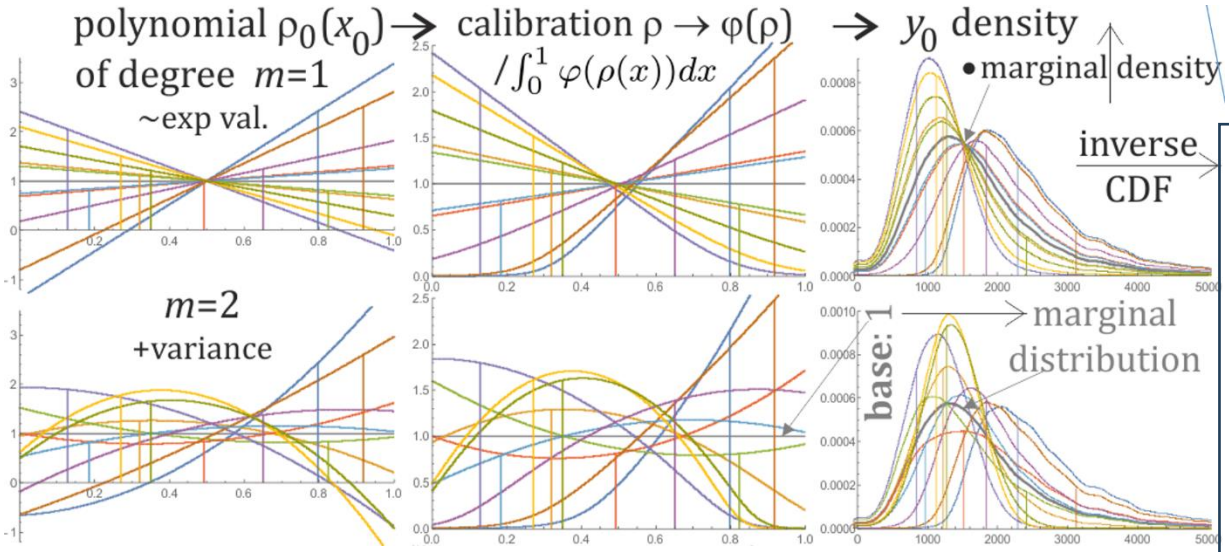
$f[x]$ reconstructed from n=1000 points: (normalized)

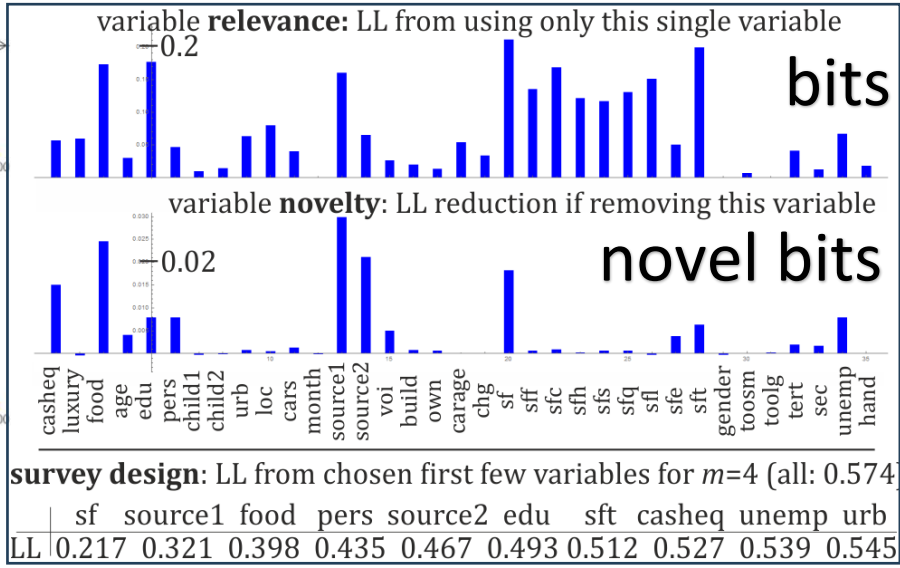


Single neuron: from joint distribution, or **direct prediction of moments** from many variables:

E.g. **linear reg.**, household income distributions:





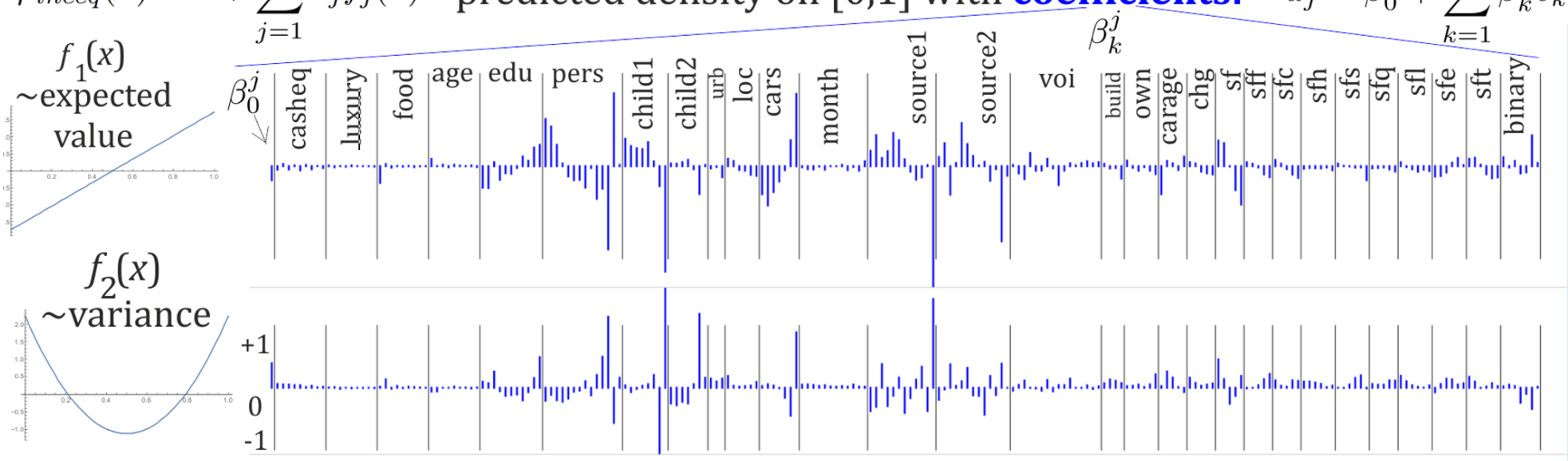


survey design: LL from chosen first few variables for m=4 (all: 0.574)

	sf	source1	food	pers	source2	edu	sft	casheq	unemp	urb
LL	0.217	0.321	0.398	0.435	0.467	0.493	0.512	0.527	0.539	0.545

$$\beta_*^j = \arg\min \sum_{(x,v)\in X} ||f_j(x) - \sum_k \beta_k^j v_k||_2^2$$

$$\rho_{inceq}(x) = 1 + \sum_{j=1}^{m} a_j f_j(x)$$ predicted density on [0,1] with **coefficients:** $$a_j = \beta_0^j + \sum_{k=1}^{222} \beta_k^j v_k$$

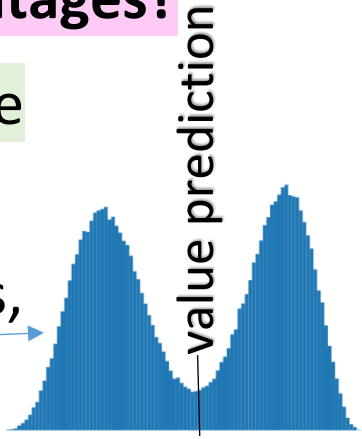


Lin reg. → NN ?

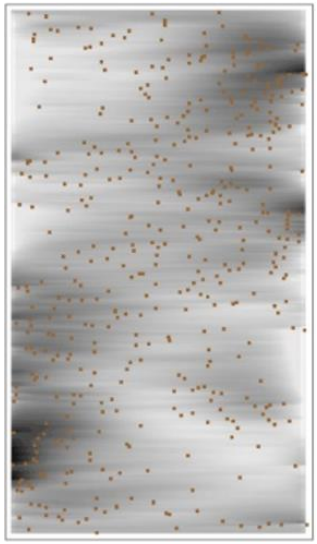
Summary of HCR articles

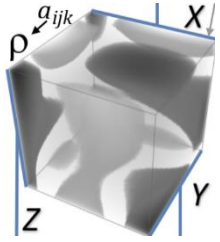
People usually focus on **prediction of values** (→ of moments → density)
where **prediction of probability distributions** gives some **advantages?**

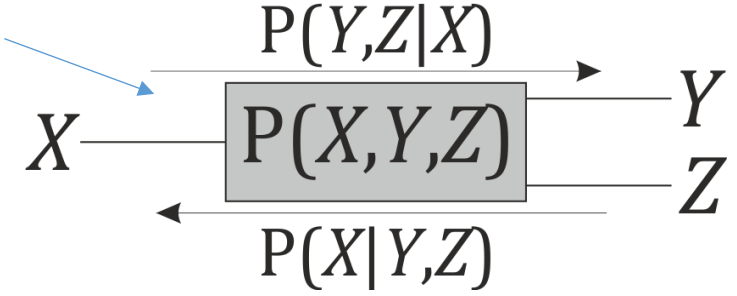
Prediction of value is of e.g. Gaussian around this value



- control of prediction **uncertainty**: variance, higher moments,
- statistical modelling in **data compression** – needs probabilities,
- controlling **multi-modality** e.g. genetic, maybe switchable?,
- proper **variable contribution evaluations** e.g. with conditional entropy,
- **Monte-Carlo**: generate random scenarios with close statistics,
- **credibility evaluation**, find outliers – low probability datapoints,
- **nonstationarity** analysis: **evolution of probability density**,
- **causality analysis, asymmetry** e.g. **multi-feature Granger**,
- **extreme statistics optimization** e.g. best batch of drugs to test,
- **uniformize** data in multiple dimensions e.g. $x' = \mathrm{CDF}_y(x)$,
- **selection** for extreme, certain values – e.g. **virtual screening** of drugs,
- **Bayes scenarios** directly from joint distribution
- **biology-inspired neural networks** …







Density propagation (using $\int_0^1 f_i(x)\ f_j(x)dx = \delta_{ij},\ \ f_0 = 1$)

$$\rho(x|y) = \sum_i f_i(x) \frac{\sum_j a_{ij}\, f_j(y)}{\sum_j a_{0j}\, f_j(y)} \quad \text{for} \quad \rho(y) = \sum_k b_k\, f_k(y) \quad \textbf{as moment vector}$$

$$\rho(x) \leftarrow \int_0^1 \rho(x|y)\ \rho(y)\, dy \approx^{\text{norm}} \int_0^1 \sum_{ijk} \left(f_i(x) a_{ij} f_j(y)\right) \left(b_k f_k(y)\right) dy =^{\text{ort}}$$

replace $f_j(y)$ with b_j

$$= \sum_{ij} f_i(x)\, a_{ij} b_j \xrightarrow[0^{\text{th}} = 1]{\text{normalize}} \sum_i f_i(x)\, \frac{\sum_j a_{ij}\, b_j}{\sum_j a_{0j} b_j}$$

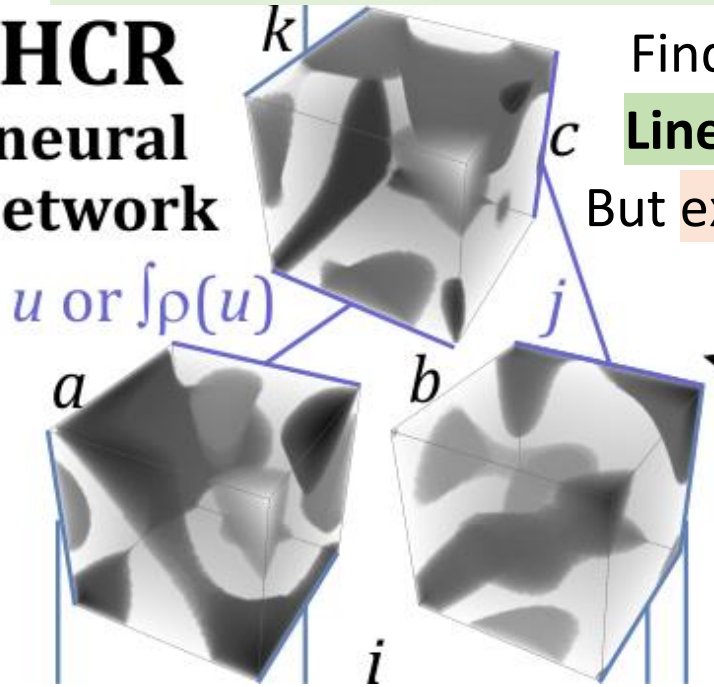
≈ constant denominator approx.

$$\int_0^1 \rho(x|y,z)\ \rho(y,z)\, dydz \approx \sum_i f_i(x)\, \frac{\sum_j a_{ijk}\, f_j(y) f_k(z)}{\sum_j a_{0jk}\, f_j(y) f_k(z)} \approx \sum_i f_i(x)\, \frac{\sum_j a_{ijk}\, b_{jk}}{\sum_j a_{0jk}\, b_{jk}}$$

Values $f_j(y), f_j(y) f_k(z) \leftrightarrow b_j, b_{jk}$ **densities** $\rho(y) = \sum_k b_k\, f_k(y)$

Approximation: directly work on NN parameters – linear, skipping normalization

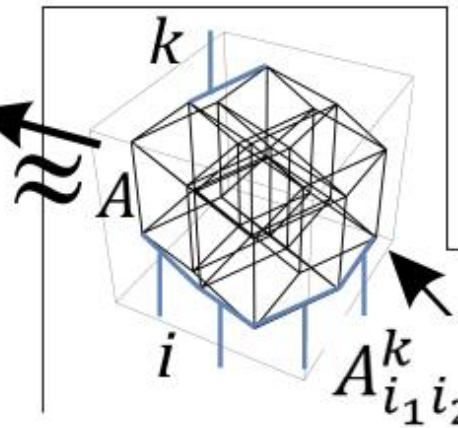
HCR neural network

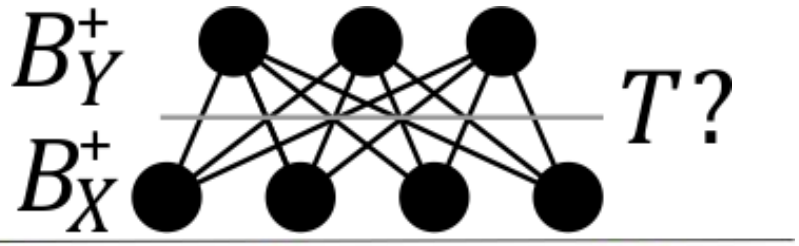


Find joint distribution as a **huge tensor** and decompose?

Linearization: nonlinearities $\{f_i(x)\}$ only in input/output

But exponential size if all dependencies – need decoupling



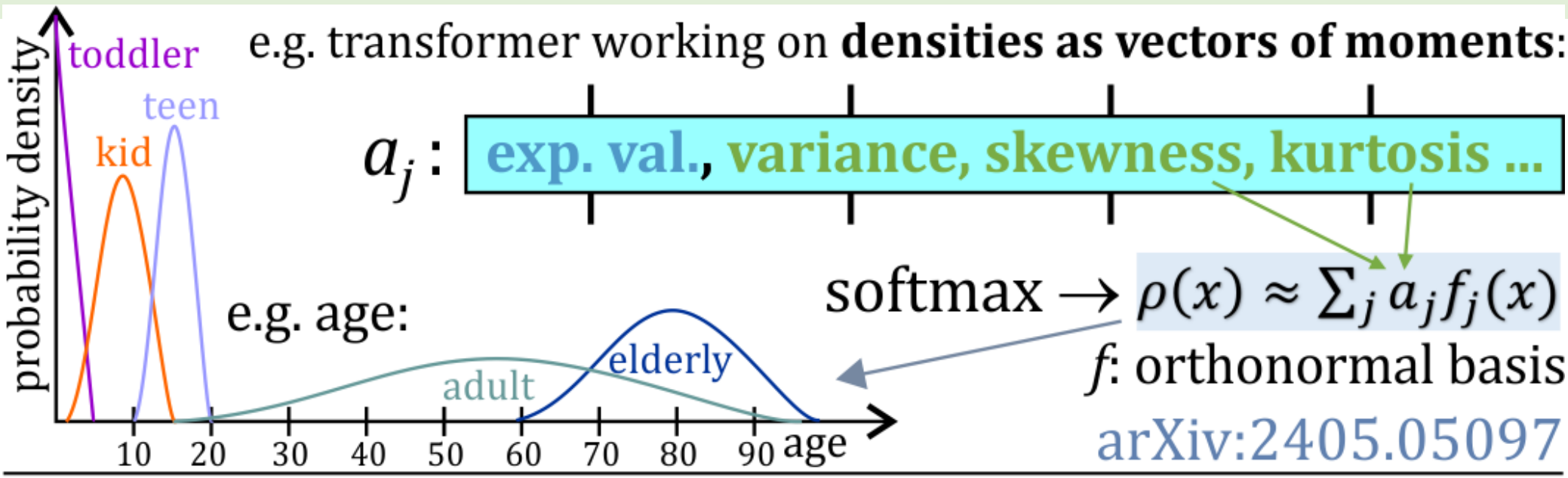


- tensor decomposition

$$A^k_{i_1 i_2 i_3 i_4} \approx \sum_{j_1, j_2} a^{j_1}_{i_1 i_2}\, b^{j_2}_{i_3 i_4}\, c^k_{j_1 j_2}$$

Density embeddings – with probability distributions of features

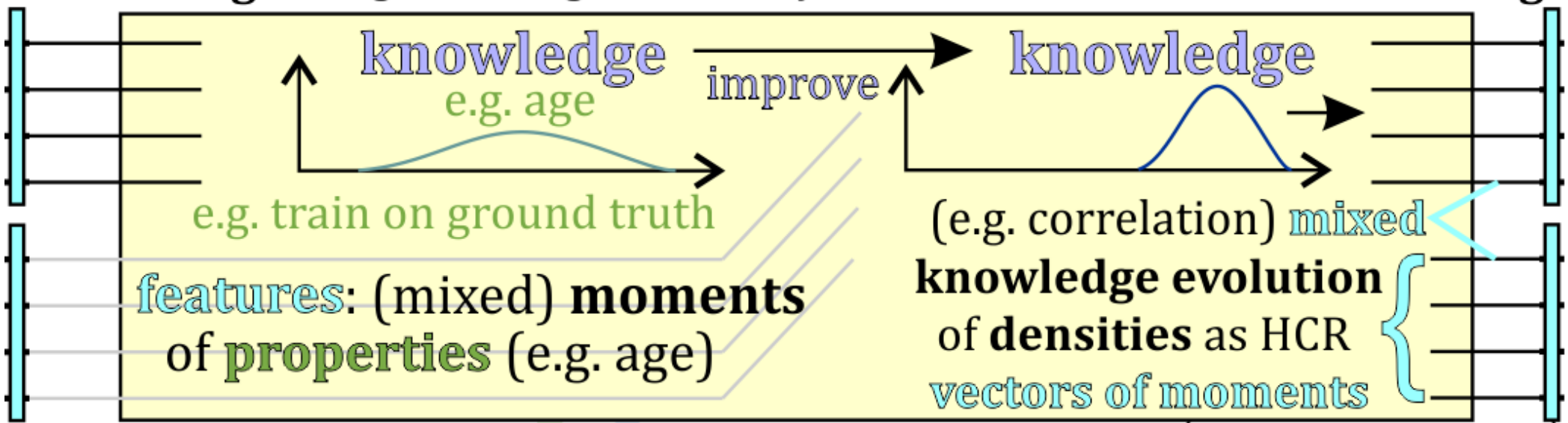
Ideally: start with population distribution → improve e.g. reducing variance



inputs **density embeddings**

NN e.g. transformer or JEPA processing **knowledge** as **probability densities**
e.g. reducing variance by new information

outputs **density un-embeddings**



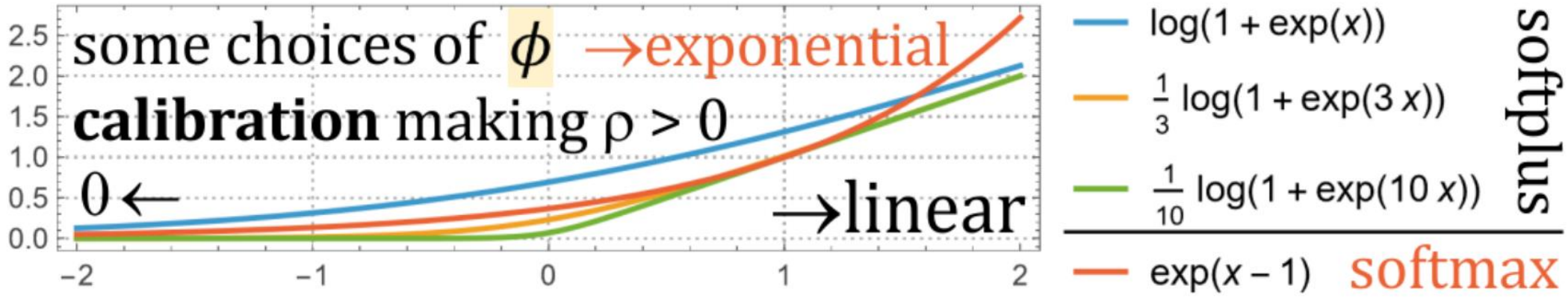
Enforce e.g. transformer to work on (joint) probability distributions?

Inputs, embedding, knowledge: $\rho(x_k) \approx \sum_{j=0}^{m} a_j f_j(x_k) \to \sum_{\boldsymbol{j}\in B} a_{\boldsymbol{j}} f_{\boldsymbol{j}}(\boldsymbol{x})$

Adding **mixed moments** e.g. correlation $a_{\boldsymbol{j}} \sim f_{\boldsymbol{j}}(\boldsymbol{x})$ becomes **softmax**

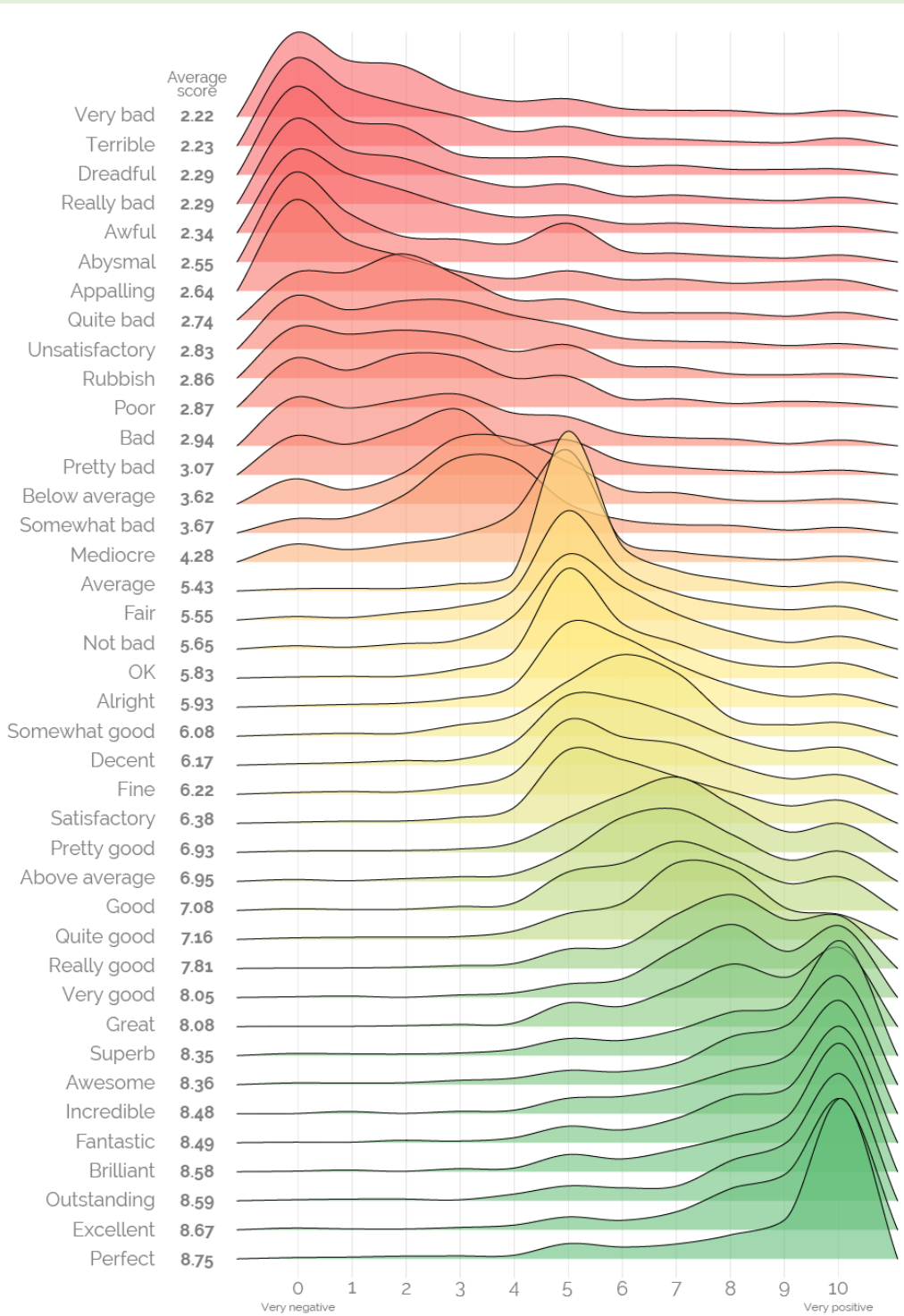
for $\varphi(\rho) = \exp(\rho - 1)$ **calibration** to $\varphi(\rho) > 0$ instead of e.g. **softplus**

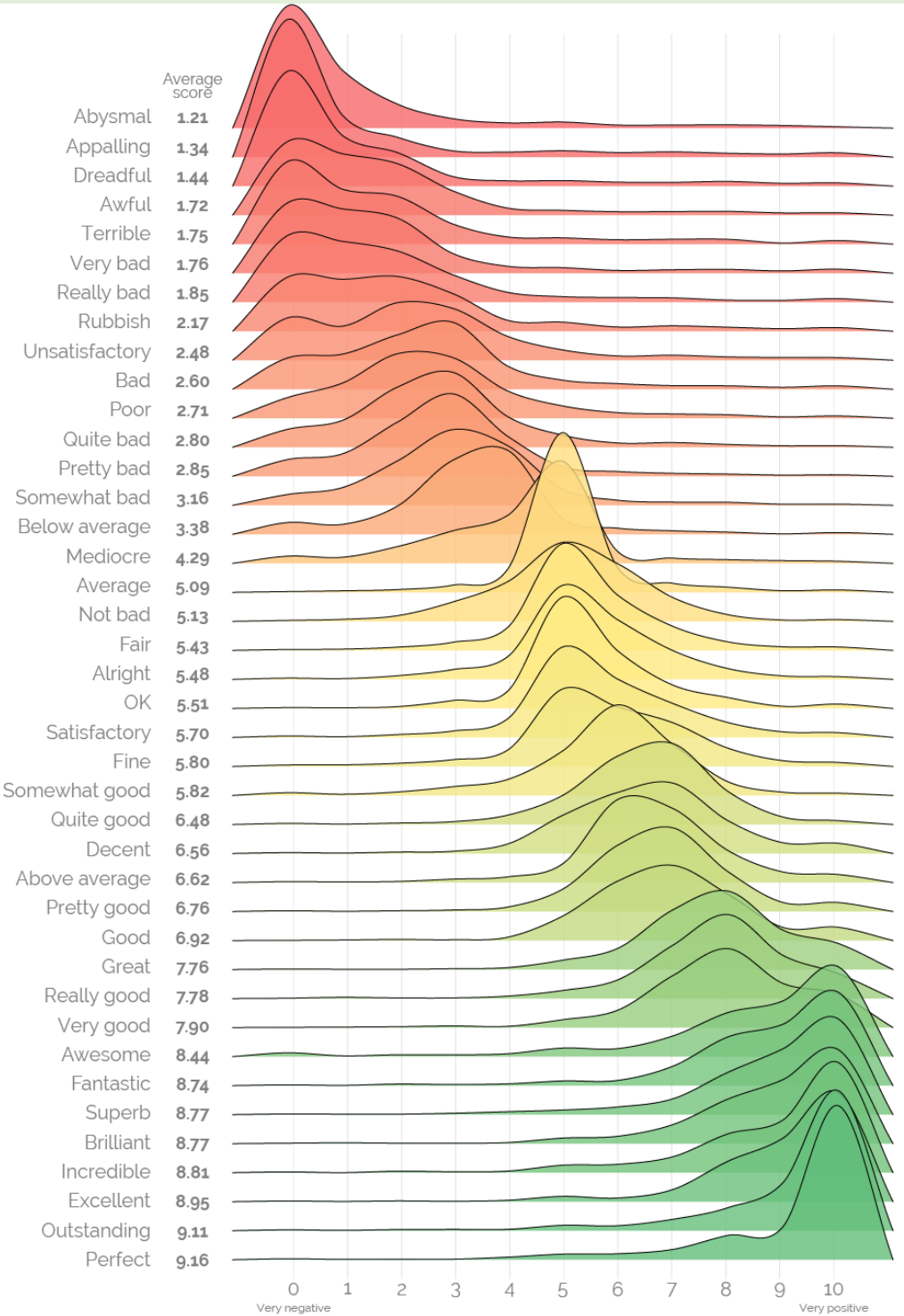
Knowledge	Output	un-embedding
Values $\{x_k\}_{k=1}^{d}$	Values $\{y_k\}_{k=1}^{d}$	Softmax $\propto \exp(\sum_k x_k y_k) = \prod_k \exp(x_k y_k)$
Values x_k Densities	Densities Values x_k	Calibrated $\propto \prod_{k=1}^{d}\left(1 + \sum_{i=1}^{m} a_{ki}\ f_i(x_k)\right)$ $\to \exp\left(\sum_k \ln\left(\varphi\left(1 + \sum_{i=1}^{m} a_{ki}\ f_i(x_k)\right)\right)\right)$
Densities $\rho_k(x)$ $= \sum_{i=1}^{m} a_{ki}\, f_i(x)$	Densities $\rho'_k(x)$ $= \sum_{i=1}^{m} a_{ki}\ f_i(x)$	Calibrated $\propto \prod_{k=1}^{d}(1 + \sum_{i=1}^{m} a_{ki}\, b_{ki})$ $\to \exp(\sum_k \ln(\varphi\ (1 + \sum_{i=1}^{m} a_{ki}\, b_{ki})))$
+mixed moments $\rho = \sum_{\boldsymbol{j}\in B} a_{\boldsymbol{j}}\, f_{\boldsymbol{j}}(\boldsymbol{x})$	+mixed moments $\rho' = \sum_{\boldsymbol{j}\in B} b_{\boldsymbol{j}}\, f_{\boldsymbol{j}}(\boldsymbol{x})$	Calibrated $\propto \left(1 + \sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}}\, b_{\boldsymbol{j}}\right)$ $\to \varphi\left(1 + \sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}}\, b_{\boldsymbol{j}}\right)$ e. g. $\exp\left(\sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}}\, b_{\boldsymbol{j}}\right)$
dens. w. mixed	Values $\boldsymbol{x} = (x_k)$ JEPA-embedding	φ e.g. $\exp\left(\sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}} \prod_{k=1}^{d} f_{j_k}(x_k)\right)$



Ground truth for embeddings as of mixed **moments**? Sentiment e.g. US vs UK queries?

Initial/fixed to speedup training (grokking)? Basis - rotation? Automatic e.g. SVD?





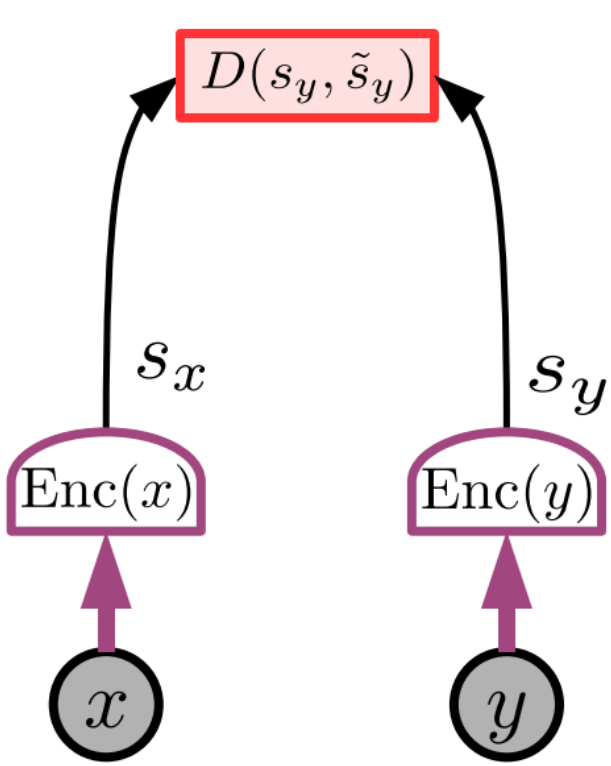
JEPA (LeCun+): Joint Embedding Predictive Architecture predict embedding, compare with distance D

HCR: predict probability density of properties (e.g. positions)

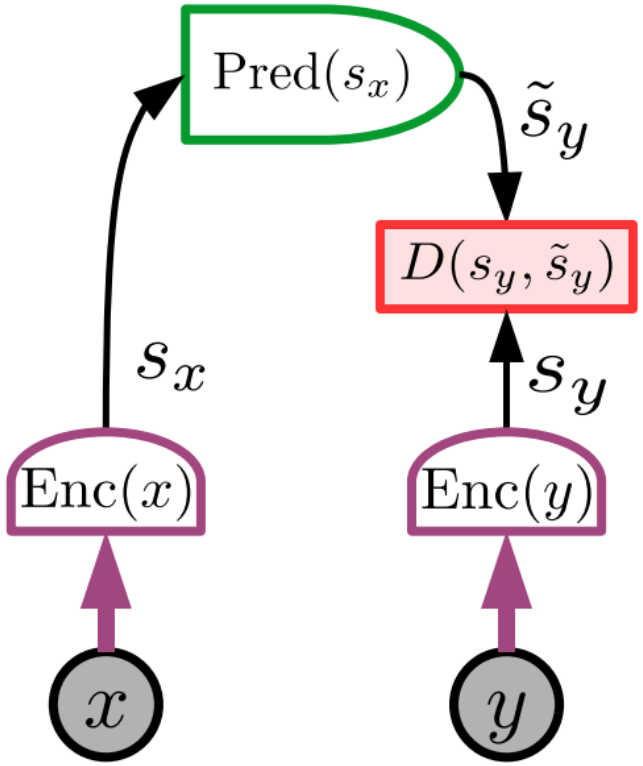
By asymmetric distance e.g. $D(\boldsymbol{a}, \boldsymbol{x}) \sim \exp\left(\sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}} \prod_k f_{j_k}(x_k)\right)$

e.g. log-likelihood, train together – including **nonlinearities of properties**

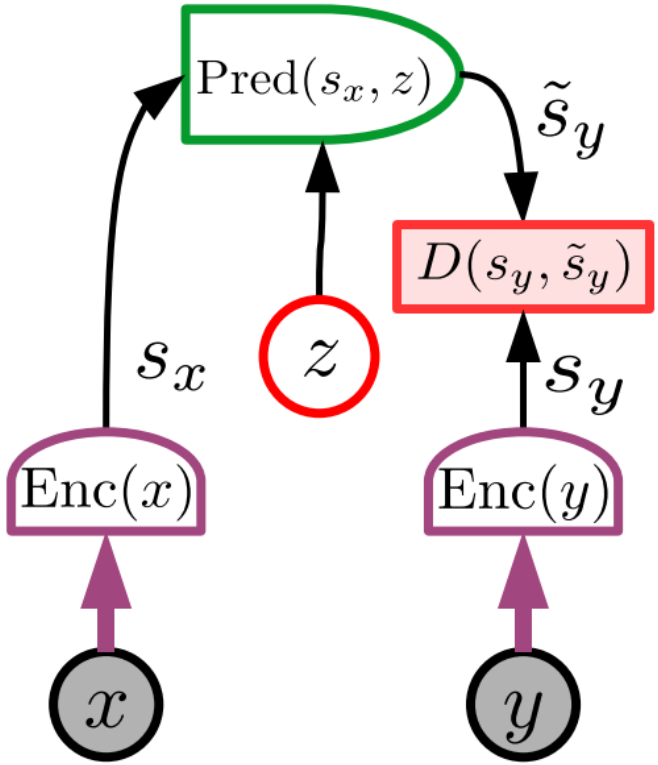
interpretability: marginal $(\#\{i: j_i > 0\} = 1)$**, pairwise** $(= 2)$**, higher**



a) Joint Embedding Architecture (JEA)
Examples: Siamese Net, Pirl, MoCo, SimCLR, BarlowTwins, VICReg,



b) Deterministic Joint Embedding Predictive Architecture (DJEPA)
Examples: BYOL, VICRegL, I-JEPA



c) Joint Embedding Predictive Architecture (JEPA)
Examples: Equivariant VICReg I-JEPA…..

Automatic calibration $\tilde{\rho} \to \varphi_\theta(\tilde{\rho}) > 0$ e.g. for log-likelihood evaluation

normalized: $\arg\max_\theta \left\{ \frac{1}{|X|} \sum_{x \in X} \ln\left(\varphi_\theta(\tilde{\rho}(x))\right) : \frac{1}{|P|} \sum_{x \in P} \varphi_\theta(\tilde{\rho}(x)) = 1 \right\}$

P- predicted e.g. random sampled $[0,1]^d$

X – dataset for log-lik.

e.g. $\varphi(v) = \alpha \max(\epsilon, v)$

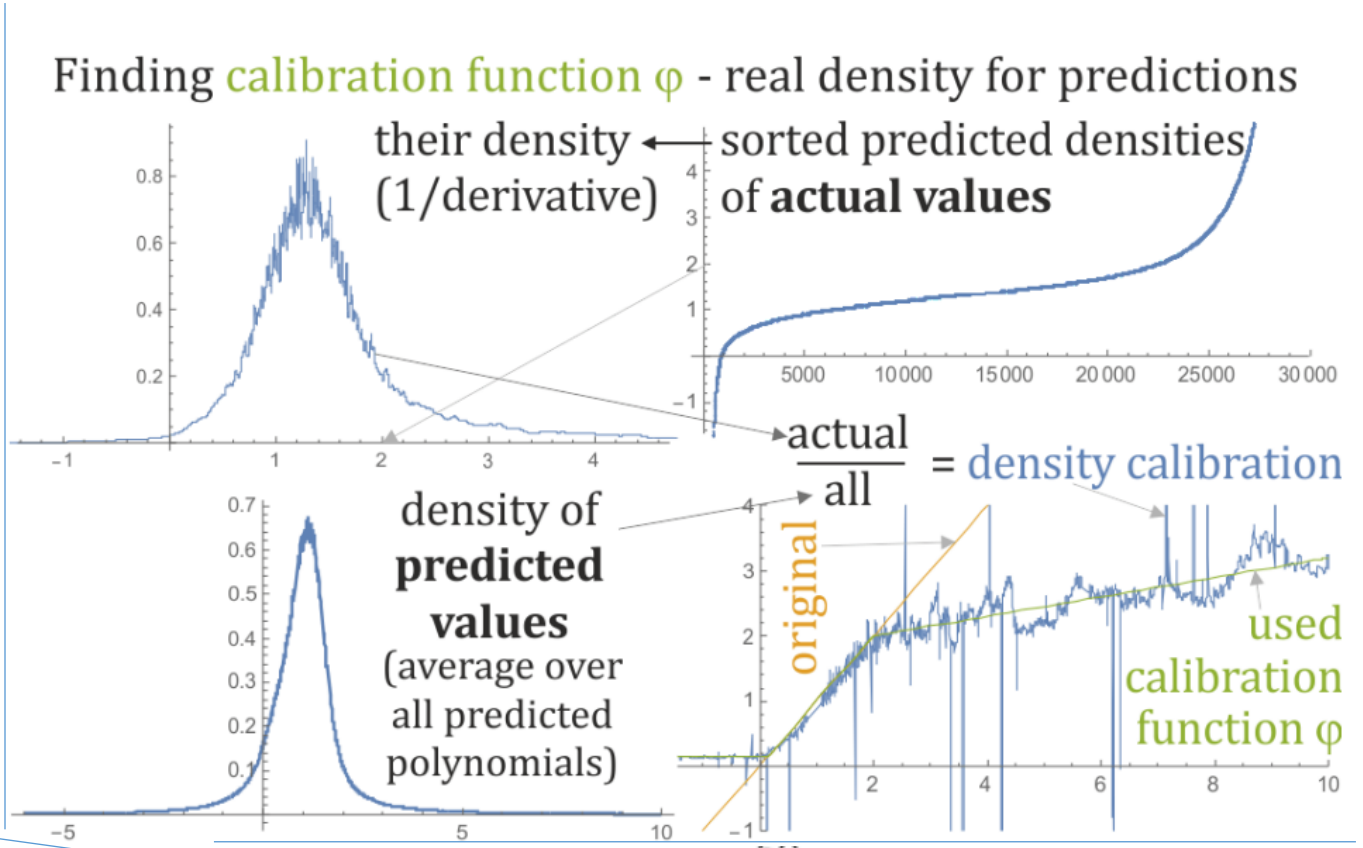
$\alpha \ln(1 + \beta \exp(\gamma v))$

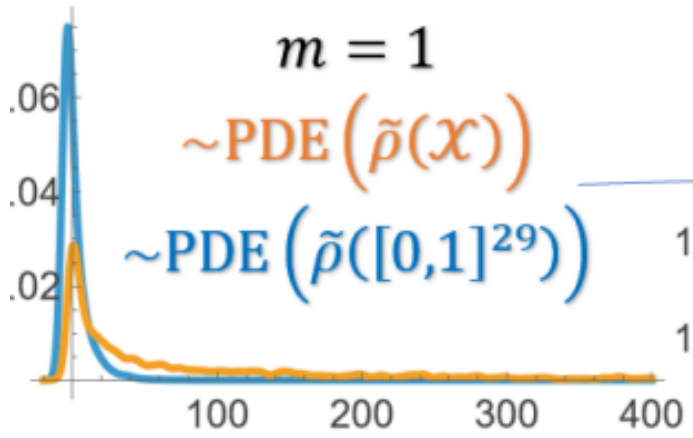
softplus … **softmax**

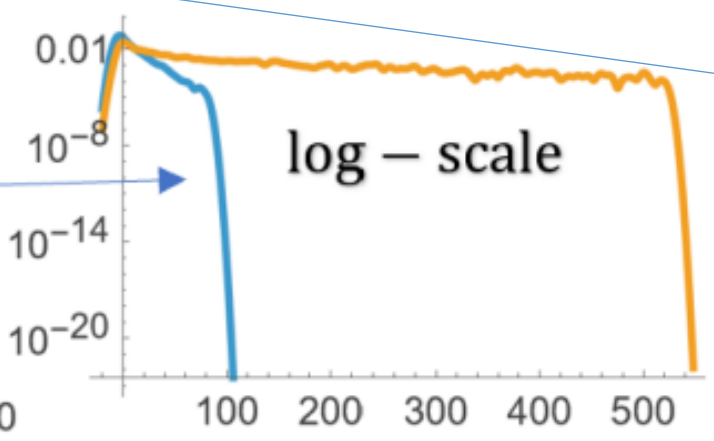
In high dimension:

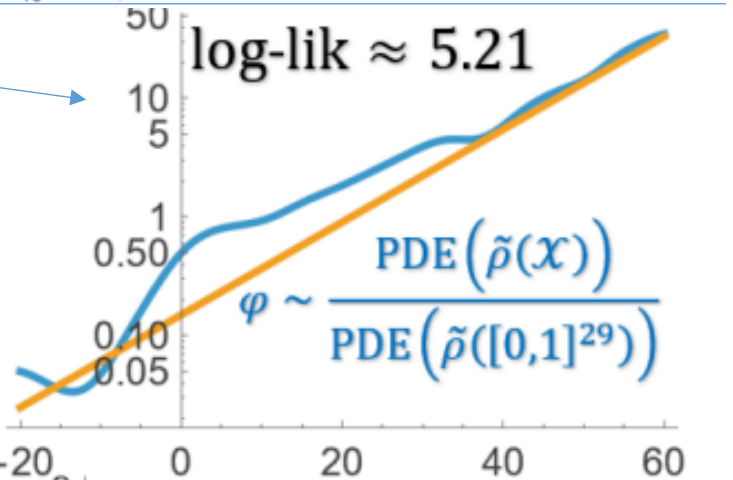
$\varphi(v) = \alpha \exp(\beta v)$

example in $[0,1]^{29}$









$d = 2$ variable (a_j) 2W neuron propagating exp. values $(i = 1)$, densities $(i \geq 1)$

Neuron containing local joint distribution model – of its connections:

(a_j) e.g. a_{ij} **matrix** $(d = 2)$ of mixed moments **as neuron parameters**

~KAN: value propagation as polynomial $E[x|y]$ → **2W** $E[y|x]$, **densities**

$$\rho(x,y) = \sum_{i,j\geq 0} a_{ij}\, f_i(x) f_j(y)$$

(a_j) neuron parameters

$$a_{ij} = \frac{1}{|\bar{X}|} \sum_{(x,y)\in\bar{X}} f_i(x)\, f_j(y)$$

direct estimation

$$\rho(x|y) = \sum_{i\geq 0} f_i(x) \frac{\sum_j a_{ij}\ f_j(y)}{\sum_j a_{0j}\ f_j(y)}$$

~KAN fit poly

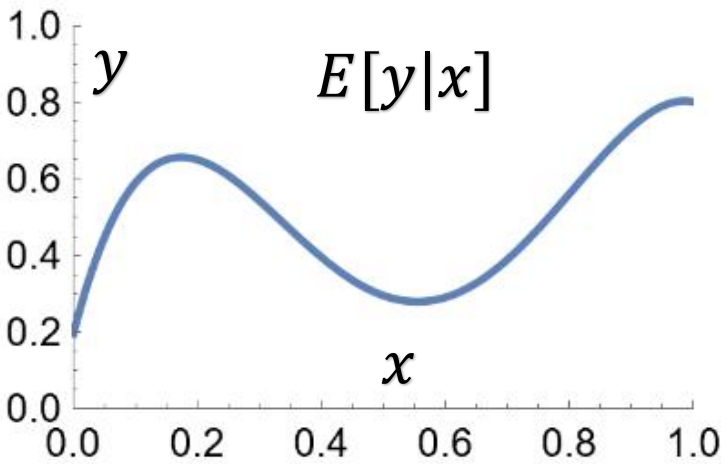
$$E[x|y] = \frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_j a_{1j}\ f_j(y)}{\sum_j a_{0j}\ f_j(y)}$$

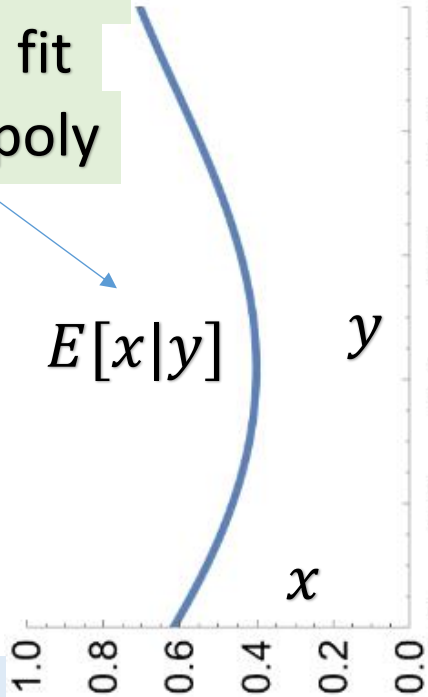
$$\rho(y|x) = \sum_{j\geq 0} f_j(x) \frac{\sum_i a_{ij}\ f_i(x)}{\sum_i a_{i0}\ f_i(x)}$$

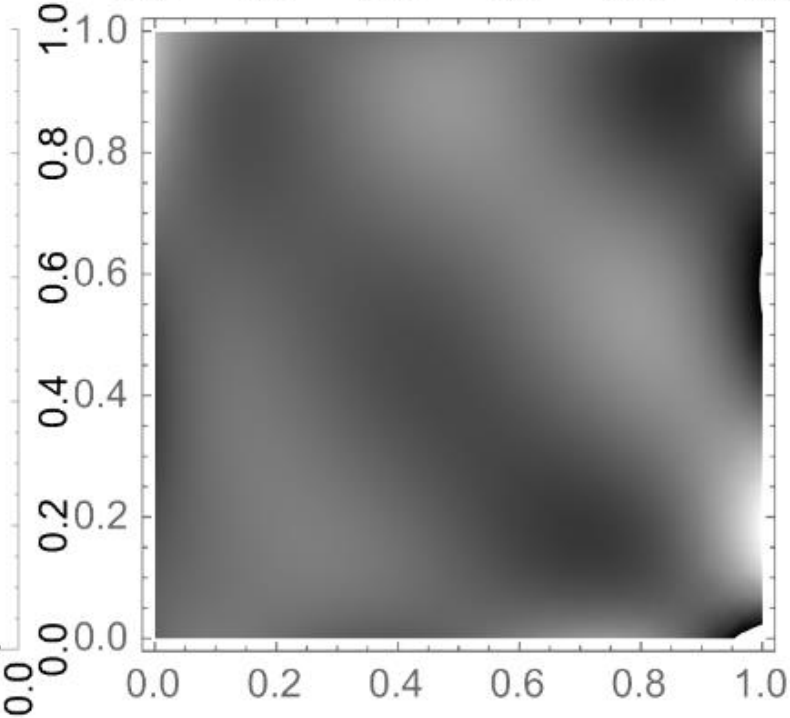
$$E[y|x] = \frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_i a_{i1}\ f_i(x)}{\sum_i a_{i0}\ f_i(x)}$$

$$I(X;Y) \approx \sum_{i,j\geq 0} (a_{ij})^2$$

mutual information estimation





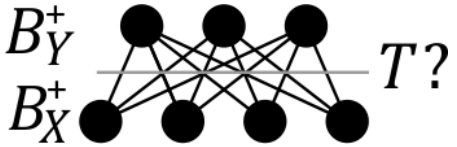


NN arXiv:2405.05097

Reduced to ~KAN for pairwise-only dependencies

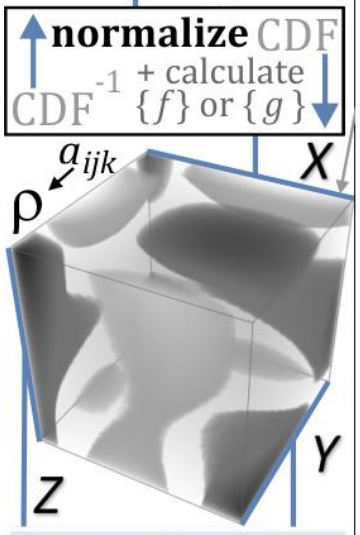
Additionally:

can extend to **triplewise** or higher, **omnidirectional propagation** ↕↔, **direct** a_j parameter **estimation**/update, propagate values or **probability distributions**, **interpretation**: moments, - cheaply calculate entropy, mutual information, additional training e.g. tensor decomposition, **information bottleneck**



d=3 variables

HCR neuron



$\rho(\boldsymbol{x}) = \sum_j a_j\, f_j(\boldsymbol{x})$

$\rho(x,y,z)$ for $d = 3$

propagation:

$\rho(x|\,y,z)$

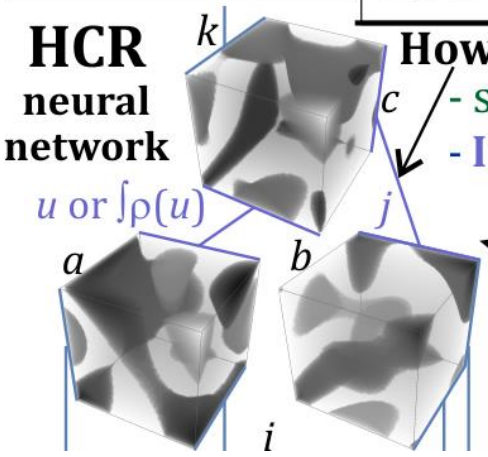
$\rho(y,z|x)$

pairwise

~KAN: $i \cdot j \cdot k = 0$

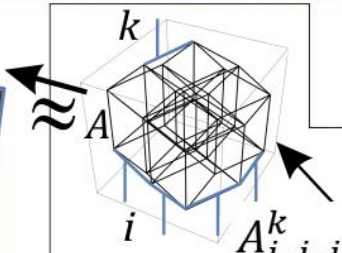
basis in [0,1]	$f_0 = 1, f_1 \propto 2x-1,\ \int_0^1 f_i(x)\, f_j(x)dx = \delta_{ij}$
↓↑ normalize	$x \leftrightarrow \mathrm{CDF}(x) \sim U[0,1]$ empirical/param.
HCR joint density	$\rho(x,y,z) = \sum_{ijk\in B}\ a_{ijk}\ f_i(x) f_j(y) f_k(z)$
static **estimation** from $\bar{X}$ dataset	mean: $a_{ijk} = \frac{1}{\lvert\bar{X}\rvert} \sum_{(x,y,z)\in\bar{X}}\ f_i(x) f_j(y) f_k(z)$
dynamic (EMA) model **update**	$a_{ijk} \xrightarrow{(x,y,z)} (1-\lambda) a_{ijk} + \lambda\, f_i(x) f_j(y) f_k(z)$
$\rho(X = x\|y,z) \approx$ ↑ conditional	$\sum_i f_i(x) \frac{\sum_{jk} a_{ijk}\, f_j(y) f_k(z)}{\sum_{jk} a_{0jk}\ f_j(y) f_k(z)}$ current / normal.
$E[X = x\|y,z] \approx$ ↑ **propagation**?	$\frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_{jk} a_{1jk}\, f_j(y) f_k(z)}{\sum_{jk} a_{0jk}\ f_j(y) f_k(z)}$ sufficient if norm.
$\rho(y,z\|x) \approx$ ↓ conditional	$\sum_{jk} f_j(y) f_k(z) \frac{\sum_i a_{ijk}\, f_i(x)}{\sum_i a_{i00}\ f_i(x)}$ current / normal.
$E[Y = y\|x] \approx$ ↓ **propagation**?	$\frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_j a_{1j0}\, f_j(y)}{\sum_j a_{0j0}\ f_j(y)}$ polyn. KAN-like sufficient if normalized
entropy, mutual information	$H(X) \approx -\sum_{j\in B_X^+} (a_j)^2$ [nits] $I(X;Y) \approx \sum_{j_x\in B_X^+} \sum_{j_y\in B_Y^+} \left(a_{(j_x,j_y)}\right)^2$
basis optimization $(y,z) \to x$ $\{f_i(x)\} \to \{g_i(x)\}$	$M_{i,jk} = a_{ijk}$ SVD: $MM^T = \sum_i \sigma_i\, \boldsymbol{v}_i \boldsymbol{v}_i^T$ $g_i(x) = \sum_j v_{ij}\ f_j(x)$ $\boldsymbol{v}_i \cdot \boldsymbol{v}_j = \delta_{ij}$ $f_i = \sum_l v_{li} g_l$ $a_{ijk} \to \sum_l v_{li}\ a_{ljk}$

HCR neural network



How to train intermediate layers/variables?

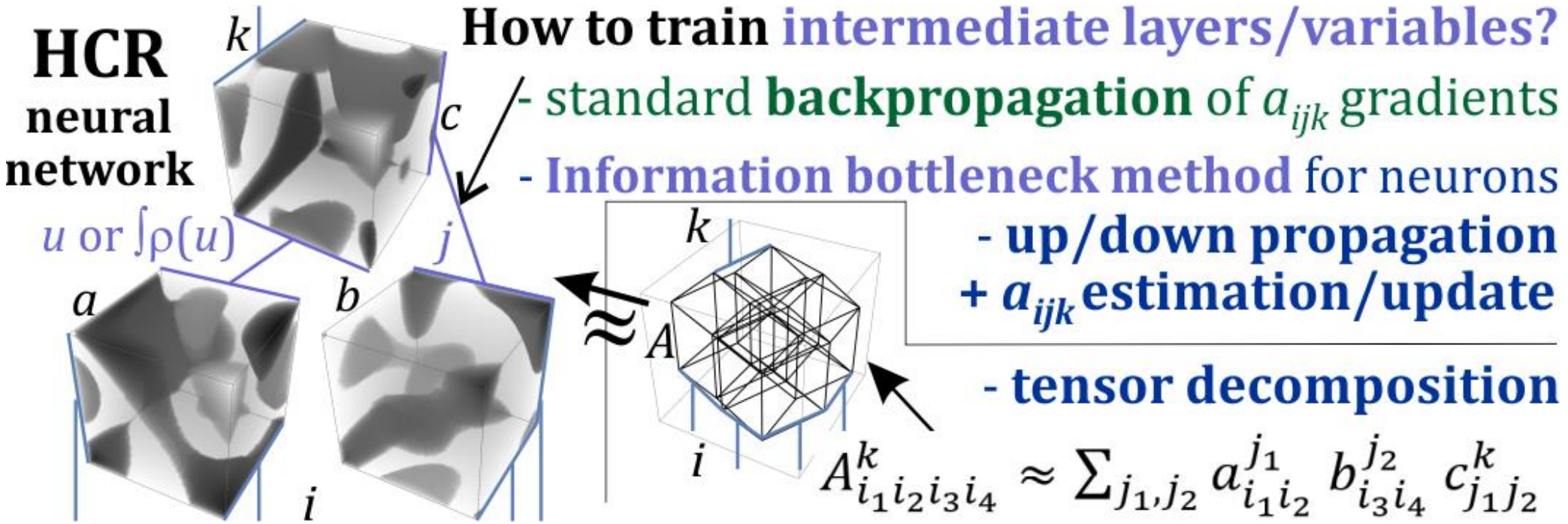
- standard **backpropagation** of a_{ijk} gradients
- **Information bottleneck method** for neurons
- **up/down propagation**
- **+ a_{ijk} estimation/update**
- **tensor decomposition**



$A^k_{i_1 i_2 i_3 i_4} \approx \sum_{j_1,j_2} a^{j_1}_{i_1 i_2}\, b^{j_2}_{i_3 i_4}\, c^k_{j_1 j_2}$

Some additional **training** approaches … biological ≠ ANN backpropagation

- Treat as parametrization, use standard **backpropagation** (KAN, MLP),
  - Use **direct estimation** e.g. $a_{ij} = \frac{1}{n}\sum_{(x,y)\in D} f_i(x)\ f_j(y)$
  - **Update parameters** e.g. $a_{ij} \to \lambda a_{ij} + (1-\lambda)\, f_i(x)\ f_j(y)$
  - **Multidirectional propagation**, e.g. up or down the layers,
    - **Tensor decomposition** approaches, starting with SVD
    - Basis optimization approaches e.g. SVD, $g_i = \sum_j v_{ij}\, f_j$
- **Information bottleneck** approach using mutual information evaluation to **directly optimize intermediate variables** $\inf_T \big(I(X;T) - \beta I(T;Y)\big)$



Can we directly train intermediate layers? (video)

Naftali Tishby, **information theoretic view** – permutation, bijection independent

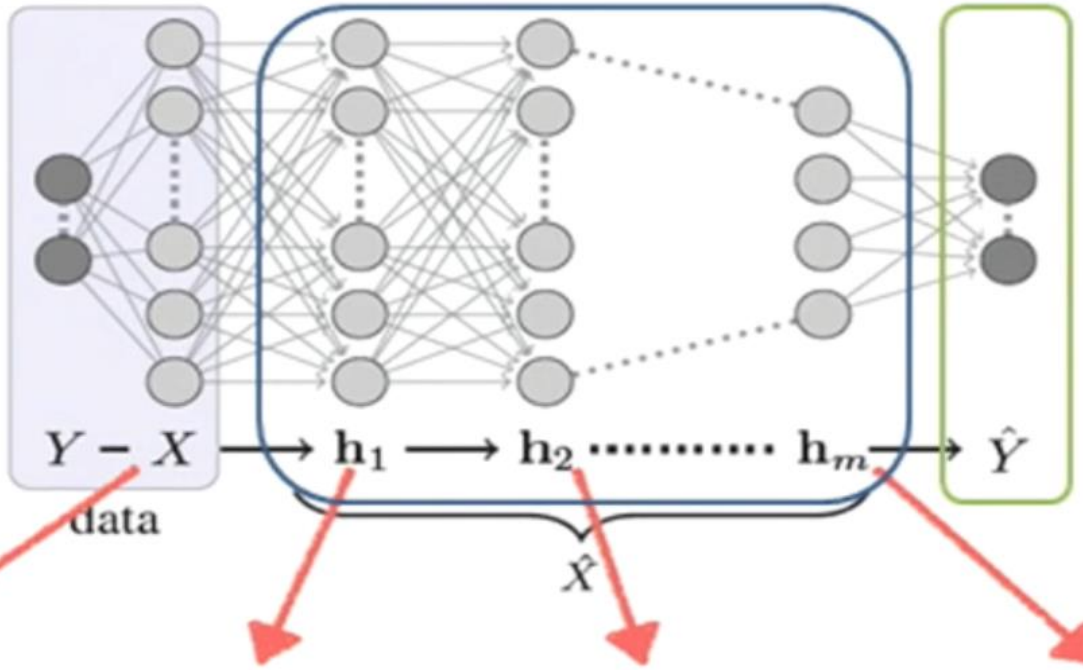
Markov process between layers, **first extract/compress essential information**

reducing **mutual information** [**bits**] $H(X) \geq I(X;T_1) \geq I(X;T_2) \geq \cdots$

Information bottleneck (Tishby): for $X \to T \to Y$ optimize $\inf_T I(X;T) - \beta I(T;Y)$

$\inf_T I(X;T)$ compression

$\sup_T I(T;Y)$ prediction



$I \sim \mathrm{HSIC} = \mathrm{Tr}(K_x, K_y)$

$K_x(x,x') = \langle \phi(x), \phi(x') \rangle$

$n \times n$ as **sample size**

ϕ Gaussian: **local, width?**

HCR: **many global** poly (f_j) **for normalized**

Information bottleneck training, **evaluation**

Tishby+ **Information bottleneck method** (2000: 4800+ citations, 2017 diagrams)

First $\max_T I(T;Y)$ prediction

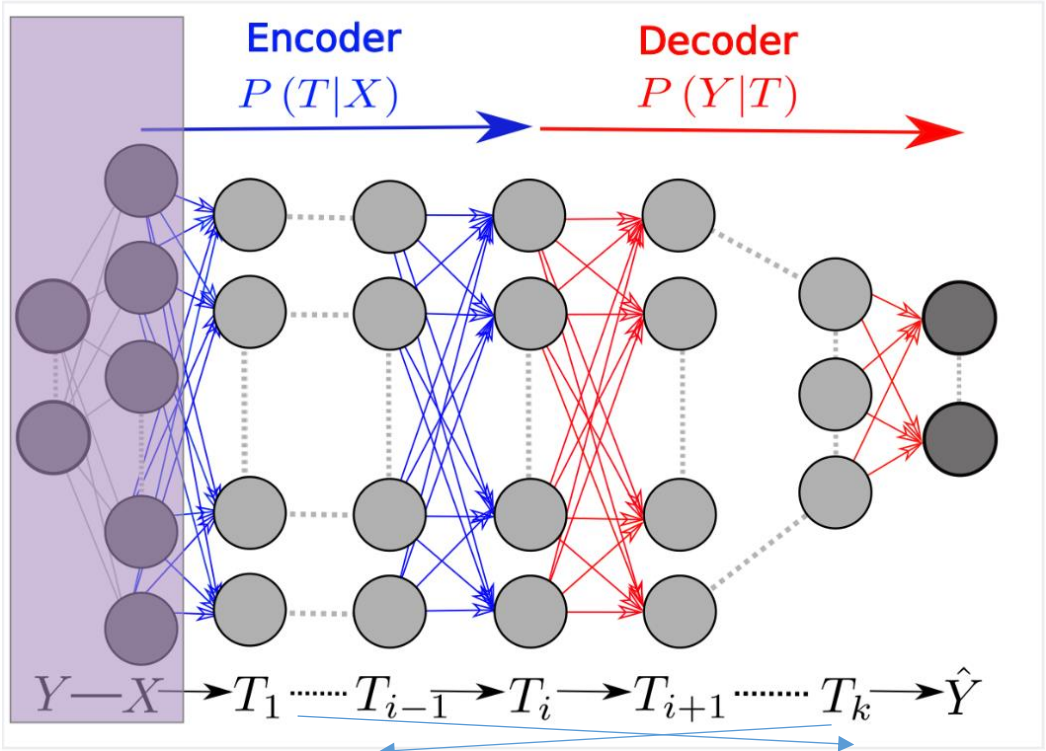
then $\min_T I(X;T)$ compression

$$\inf_{p(t|x)} \left(I(X;T) - \beta I(T;Y) \right)$$

Gaussian ~ Canonical Correlation:

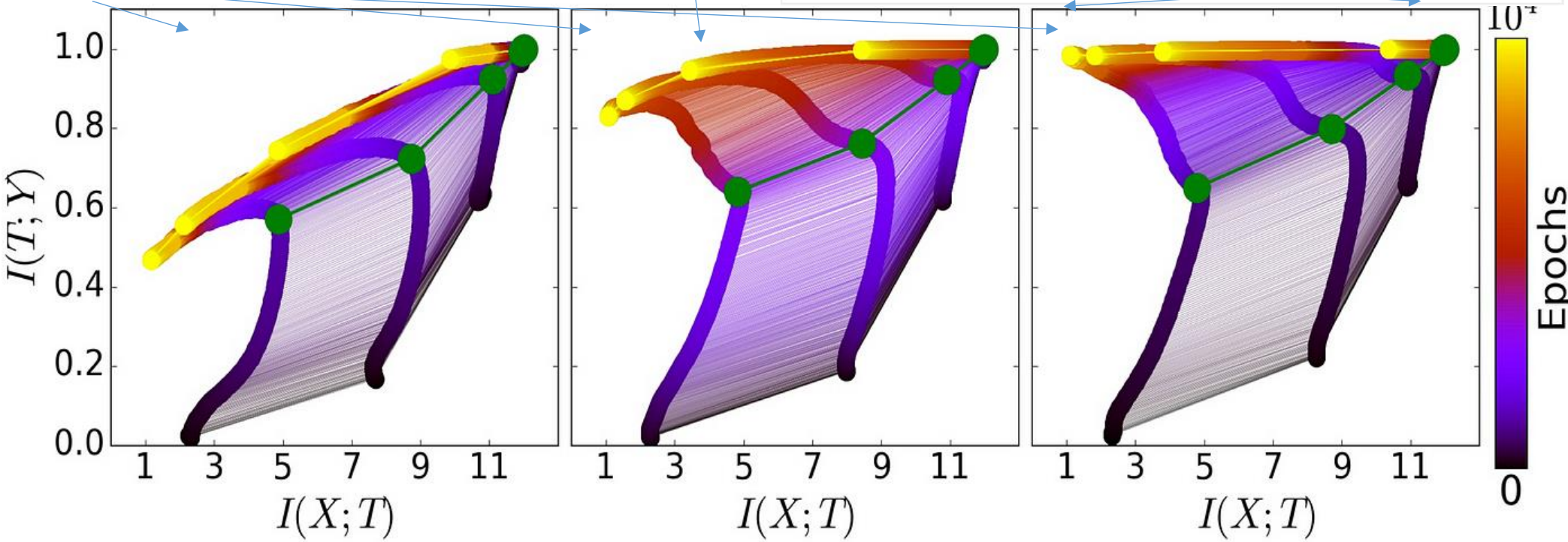
SVD of whitened covariance Σ_{XY}

… **directly minimize in HCRNN?**

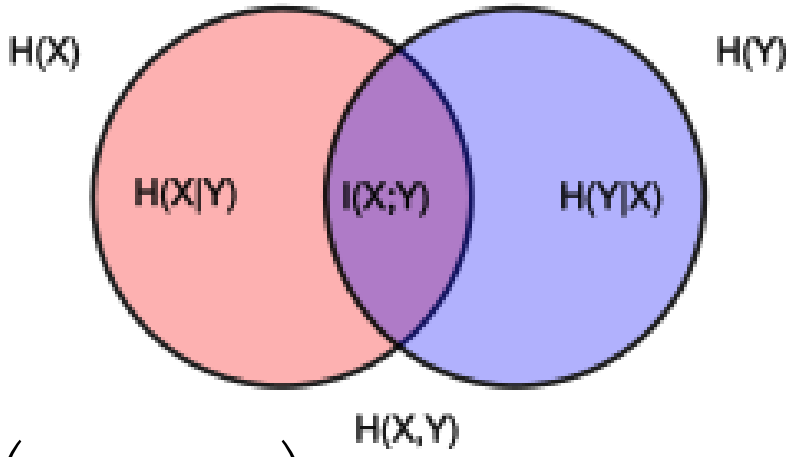


Points – different layers (video)

5%, 45%, 85% of training samples



Information theory [nats ≡ bits/ln(2)]: entropy, mutual information for HCR



$$\rho(\boldsymbol{x}) = \sum_{\boldsymbol{j}\in B} a_{\boldsymbol{j}}\, f_{\boldsymbol{j}}(\boldsymbol{x}) = 1 + \sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}}\, f_{\boldsymbol{j}}(\boldsymbol{x})$$

base; without **0**

Entropy with $\ln(1+\rho) \approx \rho + O(\rho^2)$ approx.:

$$H(X) = -\int_{[0,1]^d} \rho(\boldsymbol{x}) \ln\big(\rho(\boldsymbol{x})\big)\, d\boldsymbol{x} \approx -\int \left(\sum_{\boldsymbol{j}\in B} a_{\boldsymbol{j}}\, f_{\boldsymbol{j}}(\boldsymbol{x})\right)\left(\sum_{\boldsymbol{j}\in B^+} a_{\boldsymbol{j}}\, f_{\boldsymbol{j}}(\boldsymbol{x})\right) d\boldsymbol{x} = -\sum_{\boldsymbol{j}\in B^+} \left(a_{\boldsymbol{j}}\right)^2$$

joint: $H(X,Y) \approx -\sum_{(\boldsymbol{j}_x,\boldsymbol{j}_y)\in B_X\times B_Y\setminus\{\boldsymbol{0}\}} \left(a_{(\boldsymbol{j}_x,\boldsymbol{j}_y)}\right)^2$

conditional:

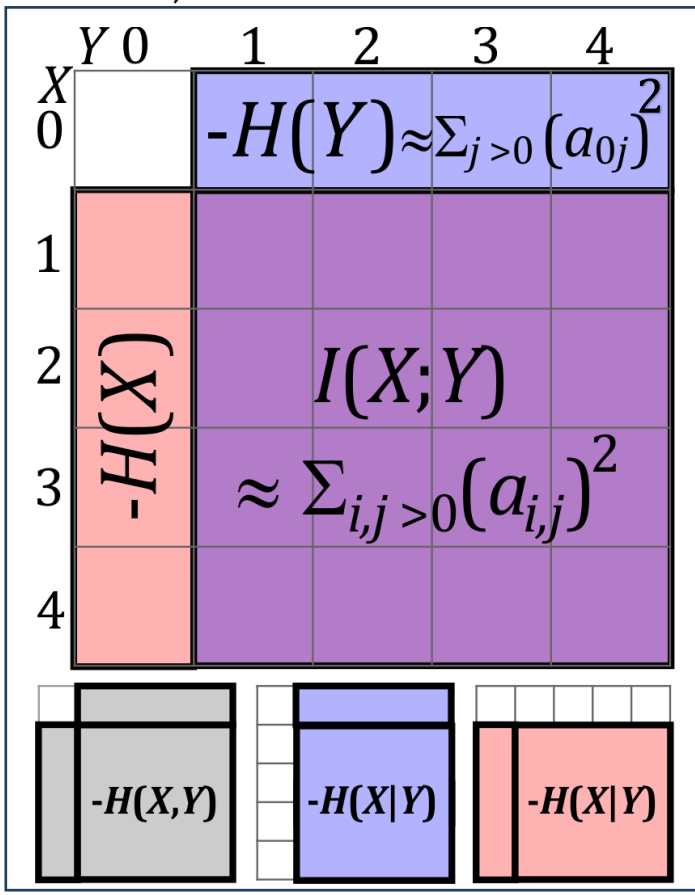
$$H(X|Y) = H(X,Y) - H(Y) \approx -\sum_{\boldsymbol{j}_x\in B_X^+} \sum_{\boldsymbol{j}_y\in B_Y} \left(a_{(\boldsymbol{j}_x,\boldsymbol{j}_y)}\right)^2$$

Mutual information: how many **bits** X says about Y

evaluation invariant to permutations, bijections

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \approx \sum_{\boldsymbol{j}_x\in B_X^+} \sum_{\boldsymbol{j}_y^+\in B_Y^+} \left(a_{(\boldsymbol{j}_x,\boldsymbol{j}_y)}\right)^2$$

nontrivial X-Y moments:



With estimation from features: $\bar{X}_{i=1..n,\,\boldsymbol{j}\in B_X^+} = \frac{f_{\boldsymbol{j}}(\boldsymbol{x}^i)}{\sqrt{n}}$

MSE:

$\boldsymbol{a} = \bar{X}^T \bar{Y}$ **getting ~HSIC:** $I(X;Y) \approx \mathrm{Tr}\big((\bar{X}\bar{X}^T)(\bar{Y}\bar{Y}^T)\big)$

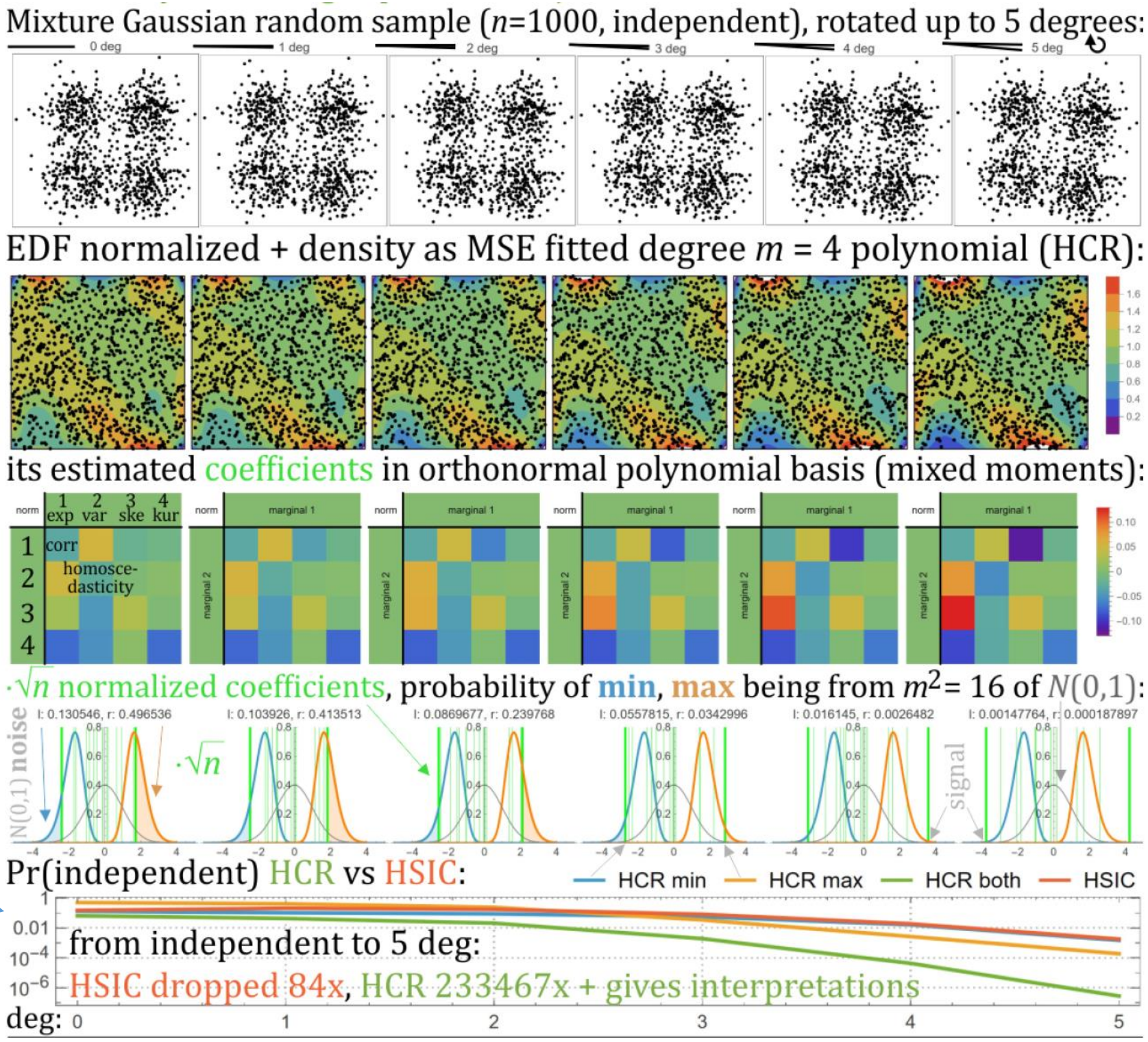
HSIC~KDE: local basis $O(n^{2.37})$, HCR: global – better generalization, cheaper $O(n|B|)$

HCR: normalize moments to $N(0,1)$ if **independent** and **test** hypothesis

Cheaper $O(|B|n)$ vs $\sim O(n^{2.37})$ SOA HSIC,

provides description as **joint distribut.:, moments**:

more sensitive



arXiv:2508.18338

Hypothesis testing are $(a_{(j_x,j_y)})$

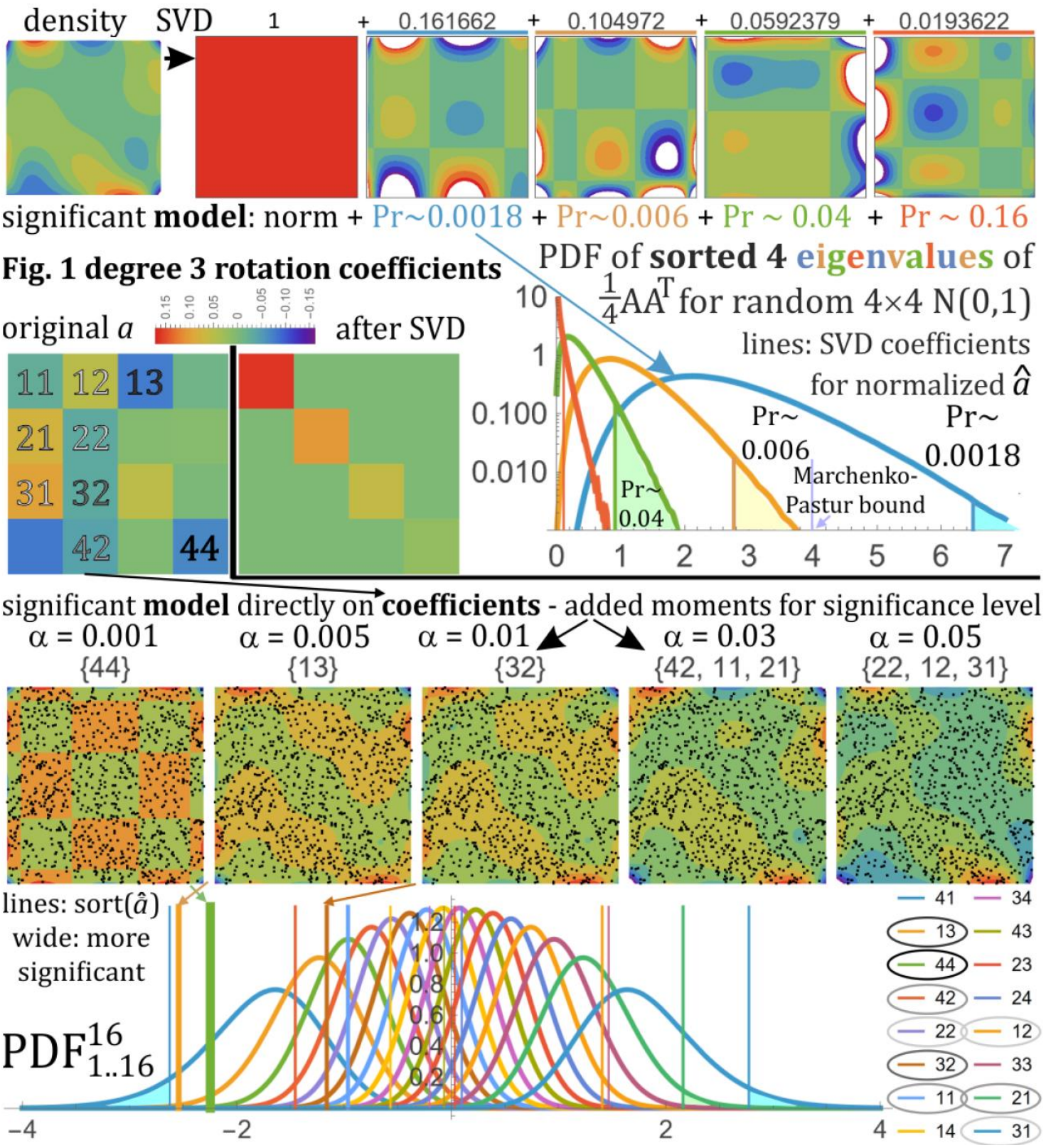
$j > 0$ moments from N(0,1)?

Compare with **min/max**, **sorted** e.g. 16 of N(0,1)?

Significant to find dependencies?

SVD and test σ? better to optimize dependencies?

~Marchenko-Pastur finite size

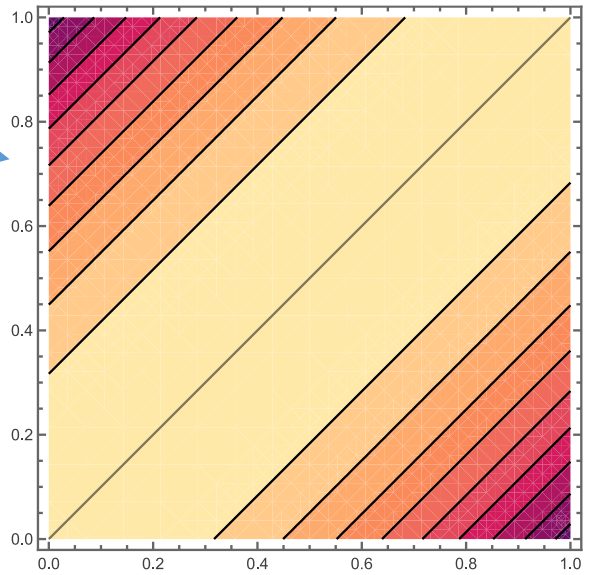


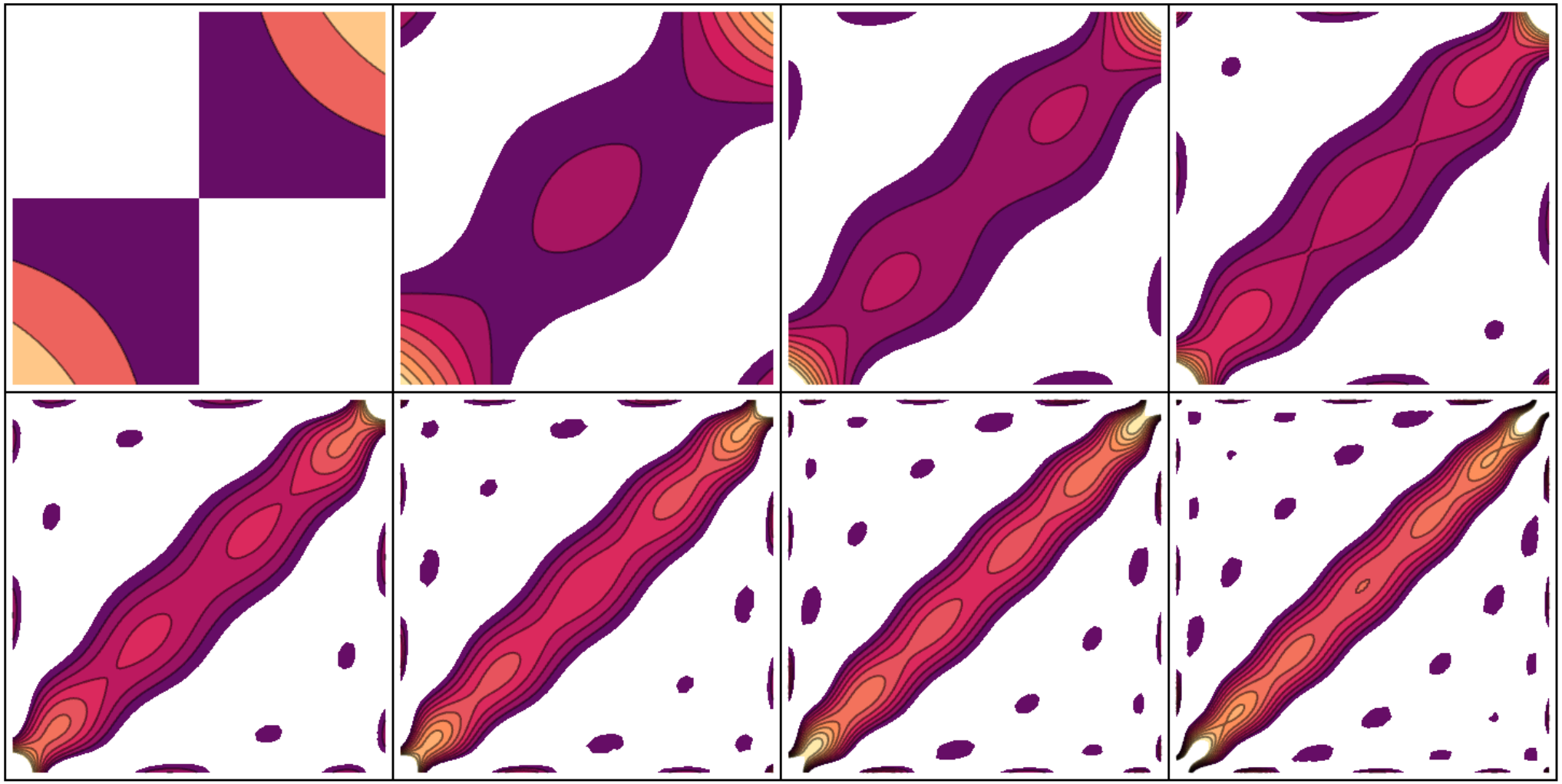
KDE/HSIC: $K_{ij} = e^{-||x_i - x_j||/\epsilon^2}$ infinite basis

HCR $C_{ij} = \sum_{k=1}^{m} f_k(x_i) f_k(x_j)$ finite basis $|B|$

For $m = 1, \ldots, 8$: drawn positive (rest negative)

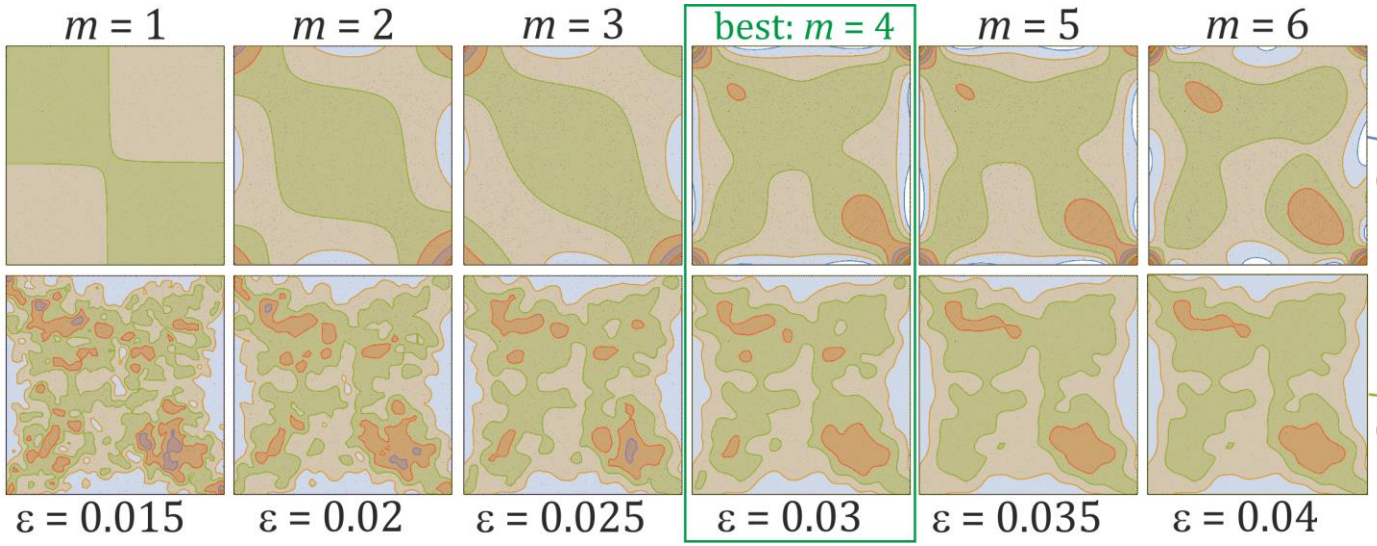
Allow $\sim O(n|B|)$ cost vs $\sim O(n^3)$ matrix product

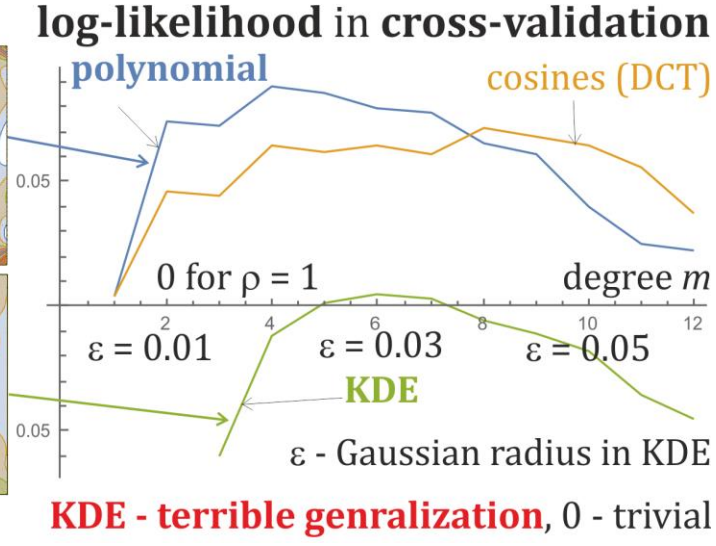




How to estimate mutual information $I(X;Y)$ e.g. for information bottleneck?

Difficult question: how many bits communicating vertical-horizontal variables:





KDE: "new points near old", HCR: find moments – better generalization

---

Kernel approach $I(X;Y) \approx \mathrm{Tr}(K_X K_y)$ for $K \in \boldsymbol{R}^{n\times n}$ needs multiplication $\sim O(n^{2.37})$

HSIC (Hilbert-Schmidt information criterion) only kernel, **used for IB: 1, 2**

Gauss $K_X = e^{-||\boldsymbol{x}^i - \boldsymbol{y}^j||/2\sigma^2}$ local ~ KDE, what σ??, not approx. of mutual information

HCR: global features: $\bar{X} = \frac{1}{\sqrt{n}}\left(f_{\boldsymbol{j}}(\boldsymbol{x}^i)\right)_{i=1..n,\boldsymbol{j}\in B_X^+}$, $\bar{Y} = \frac{1}{\sqrt{n}}\left(f_{\boldsymbol{j}}(\boldsymbol{y}^i)\right)_{i=1..n,\boldsymbol{j}\in B_Y^+}$

$O(n|B_X^+||B_Y^+|)$: $\quad I(X;Y) \approx ||\bar{X}^T\bar{Y}||_F = \mathrm{Tr}(\bar{X}^T\bar{Y}(\bar{X}^T\bar{Y})^T) = \mathrm{Tr}(C_X C_Y)$

Cost linear with sample size $\boldsymbol{n}$ ... or $\sim O(n^{2.37})$ kernel $C_X = \bar{X}\bar{X}^T, C_Y = \bar{Y}\bar{Y}^T$

HCR: better generalization, cheaper $O(n|B|)$, moments, real approx of mut. inf.

2305.09478

EEG

Multi-feature cross-correlation analysis

scatterplots

joint **density**: $\rho_{\Delta t}(y,z) \overset{\text{HCR}}{\approx} \sum_{j,k=0}^{m} f_j(y) f_k(z)\, a_{jk}(\Delta t) \overset{\text{PCA}}{\approx} 1+\sum_{i=1}^{r} f_{v_i}(y,z)\, a_i(\Delta t)$

of **EEG electrode pair** (m=10, 100+ parameters)

multi-feature delay time correlations

(r =4) dominant contributions

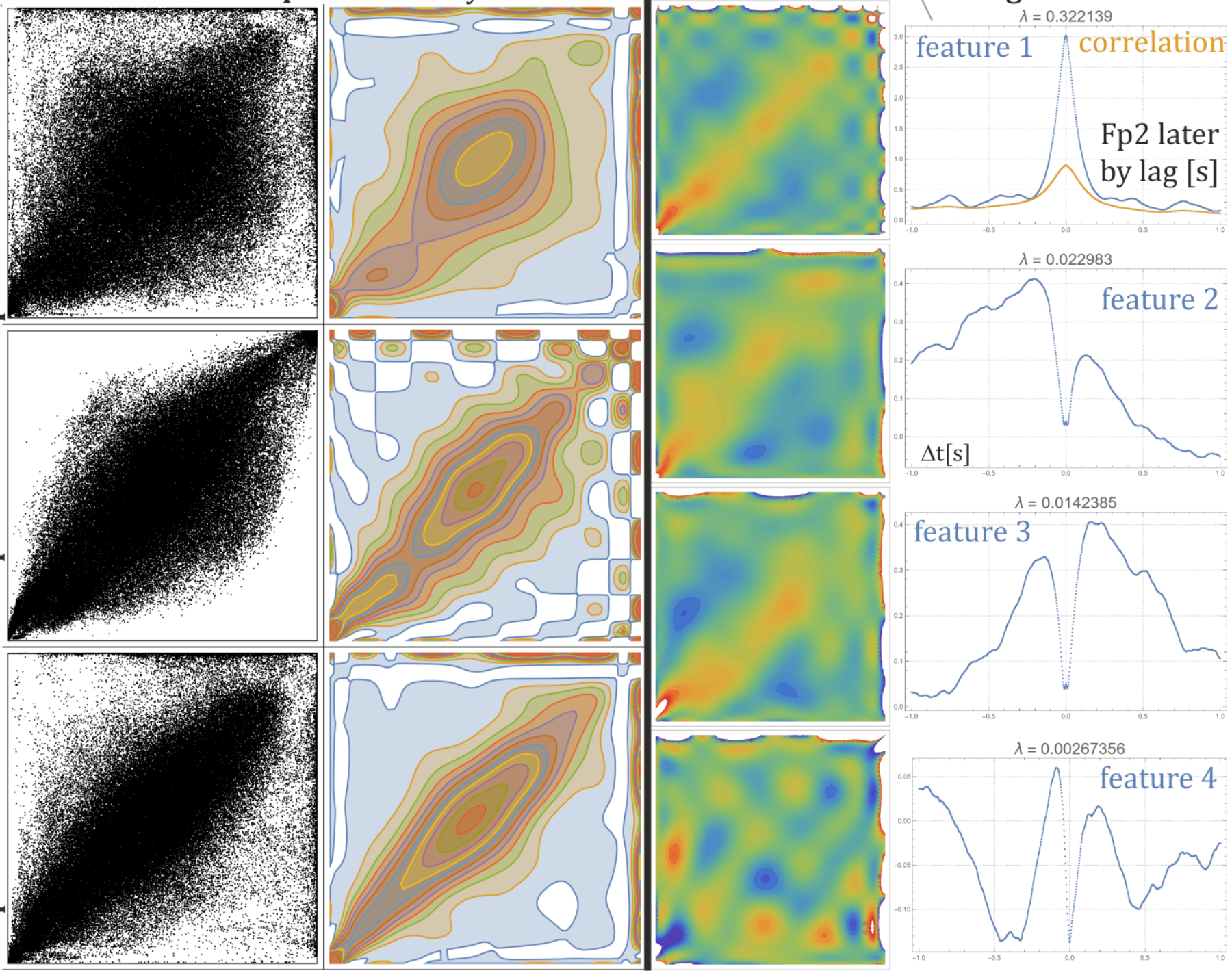
normalized values: **Fp2** → density model → dominant contr.

Δt lag-contribution

Fp1 electrode 200ms **later**

Fp1 same time

Fp1 200ms **earlier**



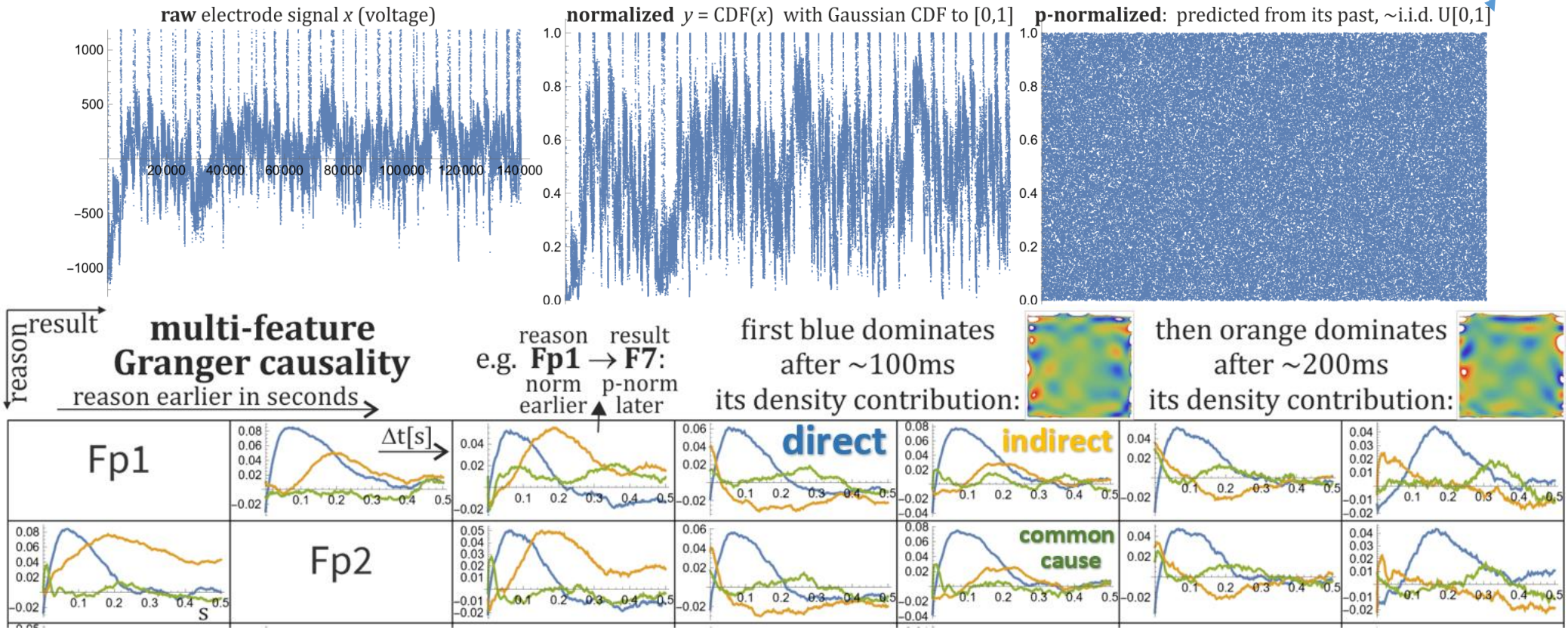
Granger causality including delay/propagation and multiple moments/joint distribution

"If a signal $\boldsymbol{Y} = (y_t)$ "Granger-causes" (or "G-causes") a signal $\boldsymbol{X}$, then **past values of $\boldsymbol{Y}$ should contain information that helps predict $\boldsymbol{X}$** above and beyond the information contained in **past values of $\boldsymbol{X}$ alone**"

Usually **true/false**: **linear regression of $\boldsymbol{X}$ with/without $(\boldsymbol{y_\tau}: \boldsymbol{\tau < t})$,**

Proposed **multi-feature Granger causality**: multiple, delay dependence:

1) **Residue** $r_t = x_t -$ "prediction of x_t from its past": $(x_\tau: \tau < t)$,
   delay Δt dependence: find correlations in $(\boldsymbol{r_t}, \boldsymbol{y_{t-\Delta t}})_{\text{all } t}$
2) **Multiple**: ~i.i.d. residue r_t, **probability prediction**: $r_t = \text{CDF}_{\tau<t}(x_t)$
   multi-feature HCR+PCA: $\rho((r_t, y_{t-\Delta t})_t) \approx 1 + \sum_{i=1}^{r} f_{v_i}(r, y)\, \boldsymbol{a_i}(\boldsymbol{\Delta t})$

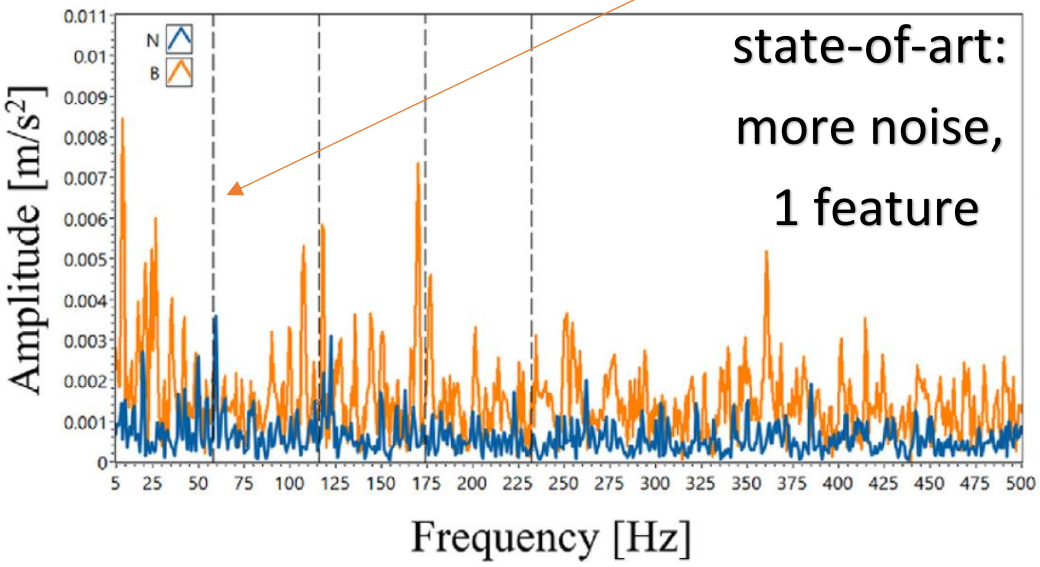


Multi-feature autocorrelation analysis + Fourier

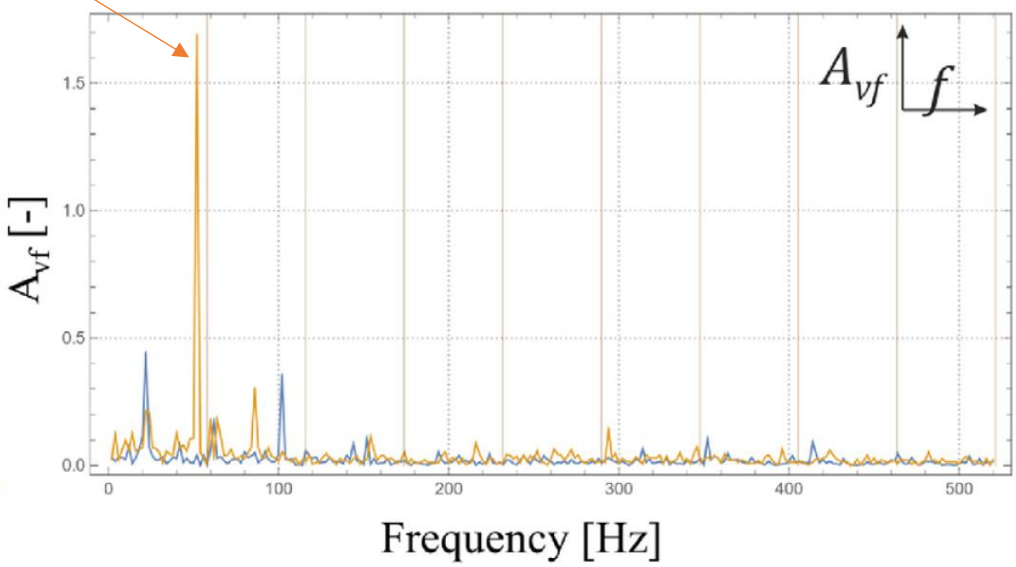
e.g. to find ~normal modes for bearing diagnostics

blue – no damage, orange – damaged (Mechanical Systems and Signal Processing)

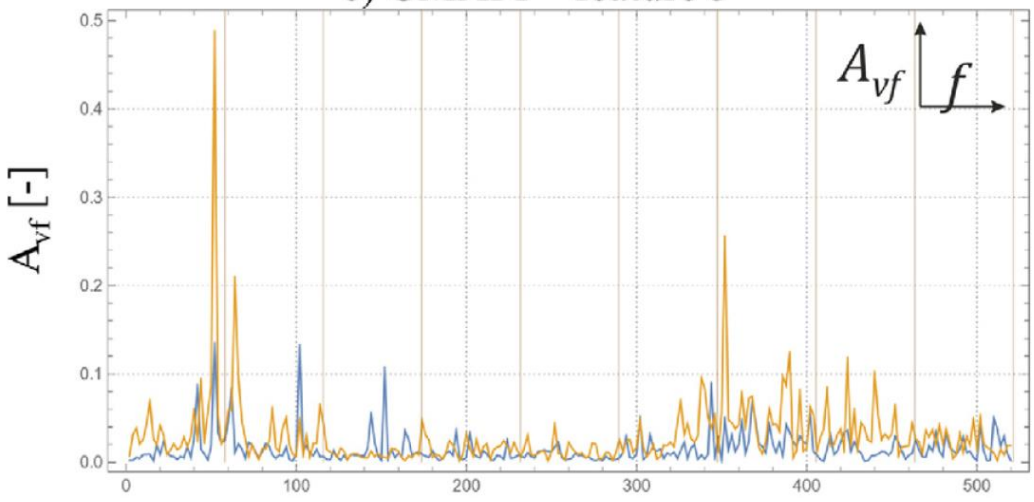
a) Envelope spectrum



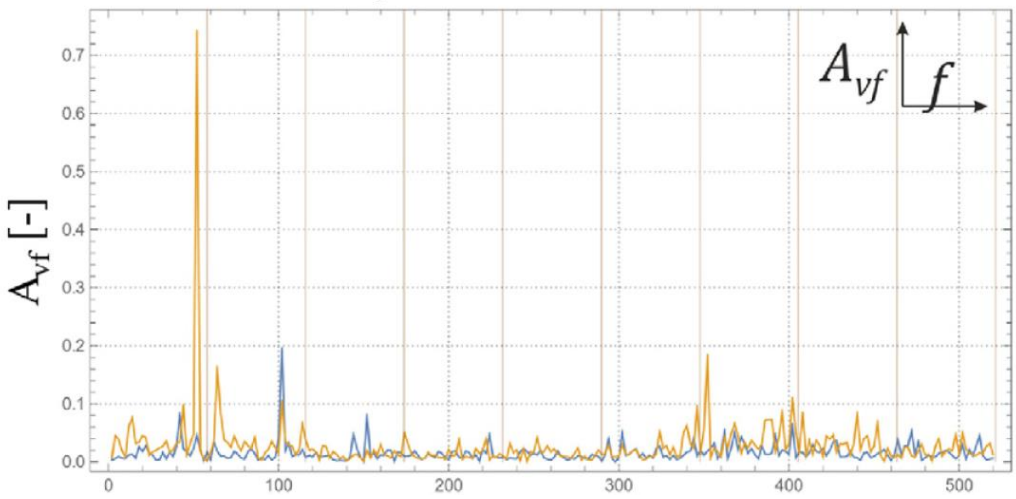
b) CMAFI – feature 2



c) CMAFI – feature 5



d) CMAFI – feature 6



evolving

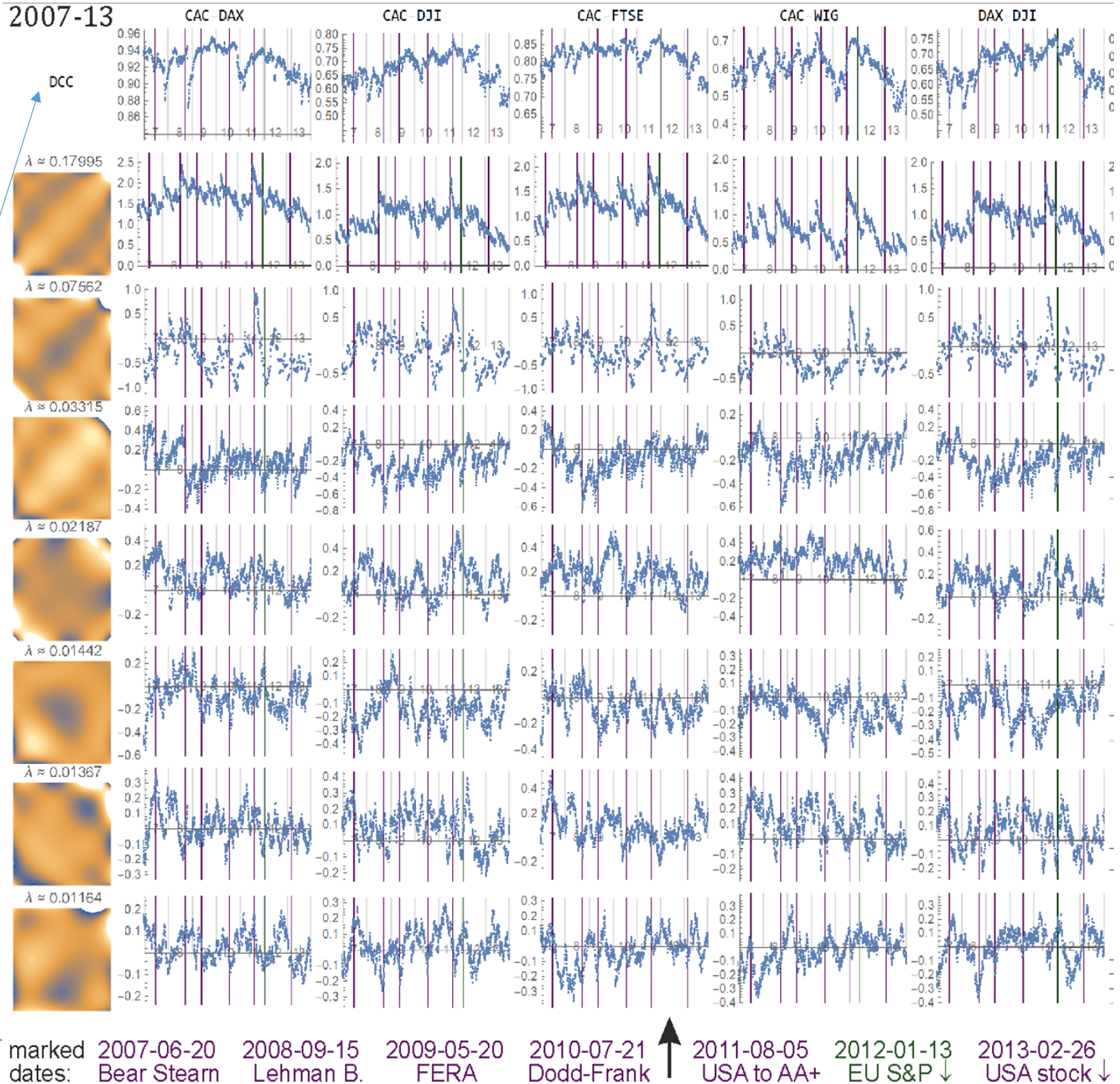
Nonsta-tionarity

Contagion bw. stocks (CEJOR)

DCC - used evolving correlation

Proposed: multiple orthogonal complementary interpretable better to detect crucial events

$$a_j \mathrel{+}= \eta\left(f_j(x_t) - a_j\right)$$

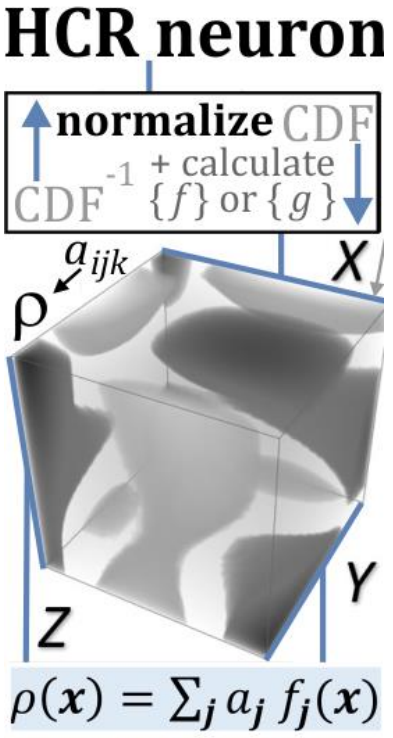


BNN learning ≠ ANN backpropagation

HCRNN: ~**KAN**-like parametrization, but with **many advantages**

- can **propagate in any direction** like **biological neural networks**,
- **propagate** values or **probability distributions**,
- **interpretation** of parameters as **mixed moments,**
- consciously add **triplewise** and higher dependencies,
- inexpensive evaluation of modeled **mutual information**,
- additional **training** approaches, e.g. **direct estimation**,

tensor decomposition, **information bottleneck** of hidden layer



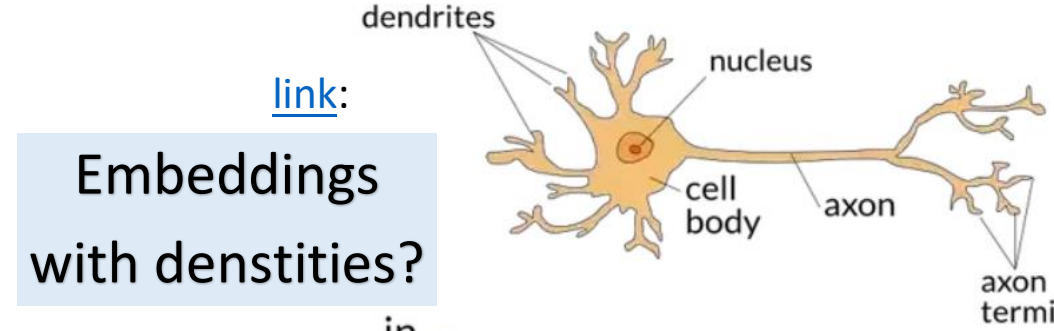
HCR joint density	$\rho(x,y,z) = \sum_{ijk\in B} a_{ijk}\ f_i(x) f_j(y) f_k(z)$
static **estimation** from $\bar{X}$ dataset	mean: $a_{ijk} = \frac{1}{\lvert\bar{X}\rvert} \sum_{(x,y,z)\in\bar{X}} f_i(x) f_j(y) f_k(z)$
dynamic (EMA) model **update**	$a_{ijk} \xrightarrow{(x,y,z)} (1-\lambda) a_{ijk} + \lambda\, f_i(x) f_j(y) f_k(z)$
$\rho(X = x\|y,z) \approx$ ↑ conditional	$\sum_i f_i(x) \frac{\sum_{jk} a_{ijk}\ f_j(y) f_k(z)}{\sum_{jk} a_{0jk}\ f_j(y) f_k(z)}$ current / normal.
$E[X = x\|y,z] \approx$ ↑ **propagation**?	$\frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_{jk} a_{1jk}\ f_j(y) f_k(z)}{\sum_{jk} a_{0jk}\ f_j(y) f_k(z)}$ sufficient if norm.
$\rho(y,z\|x) \approx$ ↓ conditional	$\sum_{jk} f_j(y) f_k(z) \frac{\sum_i a_{ijk}\ f_i(x)}{\sum_i a_{i00}\ f_i(x)}$ current / normal.
$E[Y = y\|x] \approx$ ↓ **propagation**?	$\frac{1}{2} + \frac{1}{2\sqrt{3}} \frac{\sum_j a_{1j0}\ f_j(y)}{\sum_j a_{0j0}\ f_j(y)}$ polyn. KAN-like sufficient if normalized
entropy, mutual information	$H(X) \approx -\sum_{j\in B_X^+} (a_j)^2$ [nits] $I(X;Y) \approx \sum_{j_x\in B_X^+} \sum_{j_y\in B_Y^+} \left(a_{(j_x,j_y)}\right)^2$

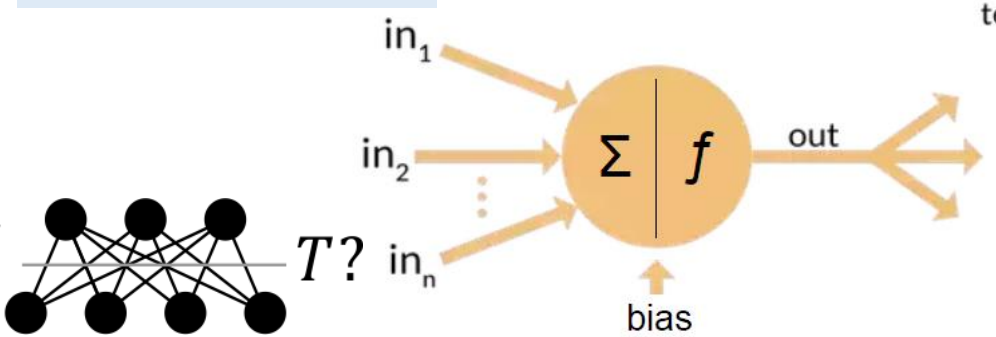
Biologically plausible?

e.g. trained 1-parameter + summation

link:

Embeddings with densities?





T?