

Enhanced Image Clustering with Random-Bond Ising Models Using LDPC Graph Representations and Nishimori Temperature Estimation

Улучшенная кластеризация изображений на спиновых моделях Изинга заданных на графах Низкоплотностных кодов с использованием температуры Нишимори

Usatyuk V., Researcher, South-West State University

Sapoznikov D., Senior Research Engineer, T8 LTD

Egorov S., Professor, South-West State University

The 8th International Conference on Deep Learning in Computational Physics



SINP MSU, Moscow, Russia

19 June, 2024



Goal of Research:

Propose a method for constructing a **priori graphs** of weights of neural networks (attention for Transformer, data tensor) with the property of unsupervised clustering on **Random-Bond Ising Models** achieve **the minimum of Free Energy** (maximum local Entropy)

Random Bond Ising Model(RBIM)

Квантовые модели Марковского случайного поля на графе

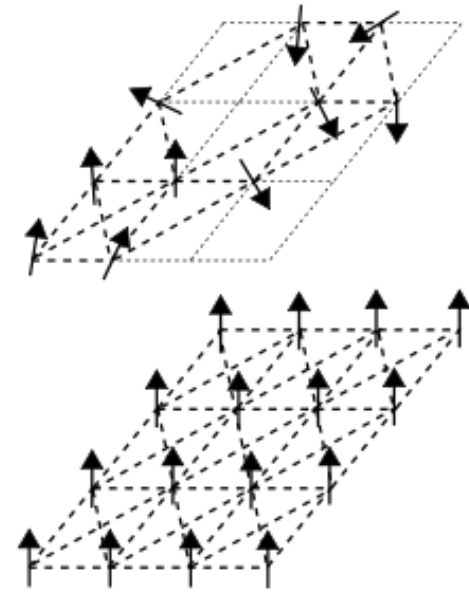
Random Bond Ising Model (RBIM) - 2D Ising model on Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$$\mathcal{H}(\sigma) = - \sum_{(i,j) \in \mathcal{E}} J_{ij} \sigma_i \sigma_j,$$

Each node $i \in \mathcal{V}$ is associated with a spin variable, $\sigma_i \in \{-1, +1\}$, \mathcal{E} - is the set of edges (bonds) J_{ij} - bond strengths, which are rand. assigned according to a specified probability distribution. In the random bond Ising model, J_{ij} can take different values, typically following a distribution such as

$$P(J_{ij}) = p\delta(J_{ij} - J) + (1 - p)\delta(J_{ij} + J),$$

where δ is the Dirac delta function, J is a positive constant representing the bond strength, and p is the probability of a bond being ferromagnetic ($J_{ij} = J$) versus antiferromagnetic ($J_{ij} = -J$).

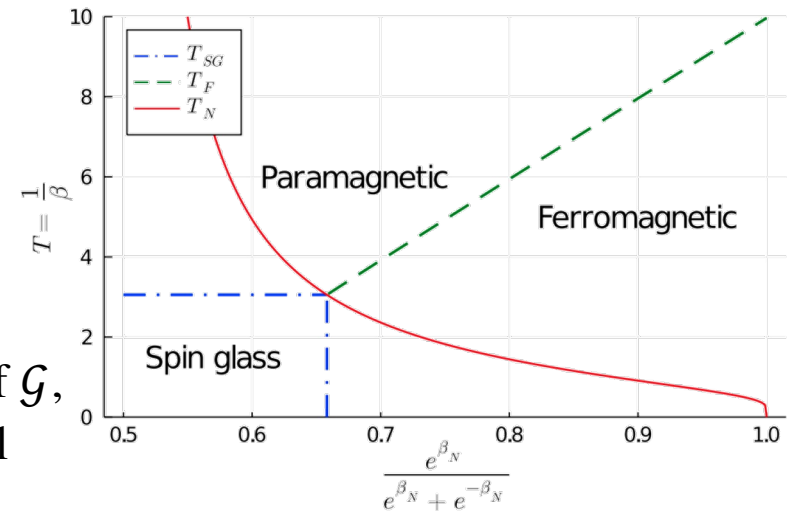
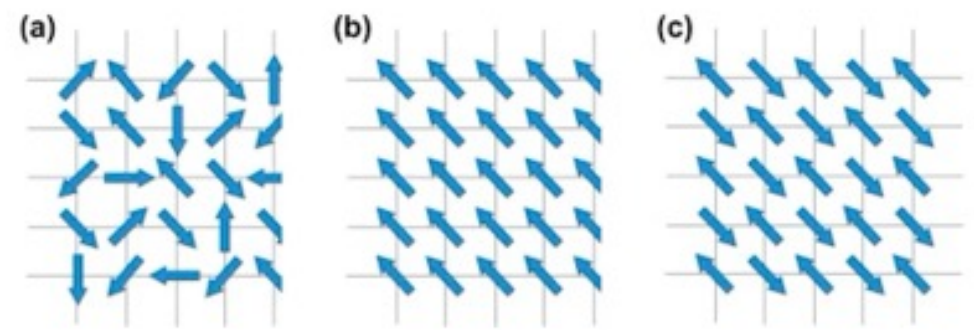
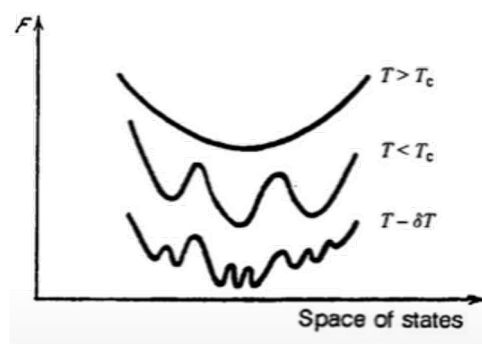
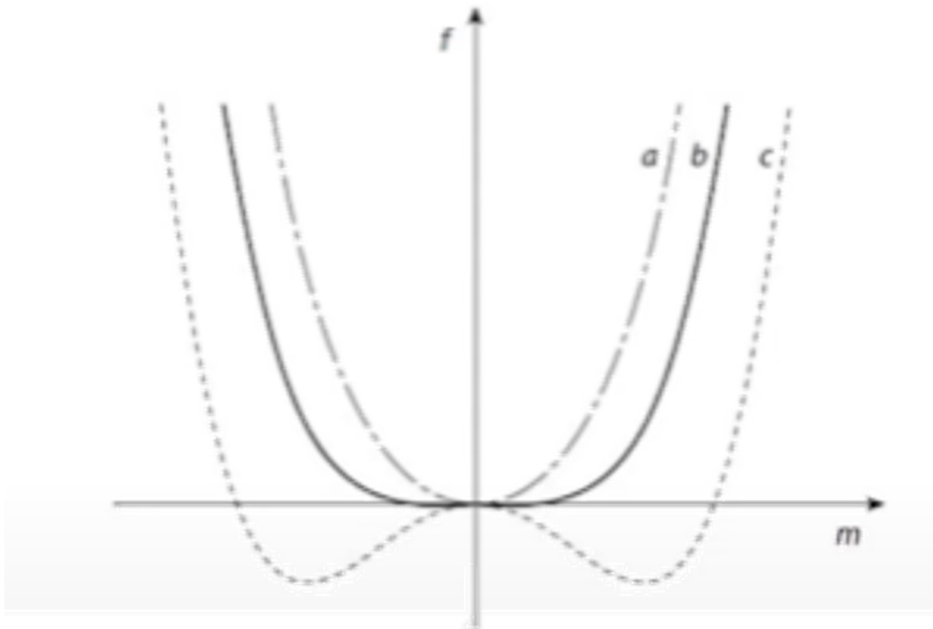


The partition function Z of the system, which is used to study the statistical properties, is given by

$$Z = \sum_{\sigma} e^{-\beta \mathcal{H}(\sigma)},$$

where $\beta = \frac{1}{k_B T}$ is the inverse temperature, with k_B being the Boltzmann constant and T the temperature.

RBIM phase transaction under different temperature



At different temperature T we can see a phase transition

Under Nishimori Temp. Let $A \in \{0,1\}^{n \times n}$ be the symmetric adjacency matrix of \mathcal{G} , defined by $A_{ij} = 1$ if $(ij) \in \mathcal{E}$, and $A_{ij} = 0$ otherwise. Let $D \in \mathbb{N}^{n \times n}$ be the diagonal degree matrix $D = \text{diag}(A\mathbf{1}_n)$.

$$\mathbb{E}[H_{\beta_N, J}] = I_n + \mathbb{E} \left[\frac{\text{th}(\beta J_{ij})}{1 - \text{th}^2(\beta J_{ij})} \right] (D - A).$$

A – non diagonal element completely defined by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

Viktor Dotsenko, Introduction to the Replica Theory of Disordered Statistical Systems *Landau Institute for Theoretical Physics, Moscow, 2001, 219 p*

Mezard M. Parisi G., Virasoro M. *Spin Glass Theory and Beyond An Introduction to the Replica Method and Its Applications. 1986, 476 p.*

H. Nishimori, J. Phys. C: Solid State Phys. 13(21), 4071 (1980), H. Nishimori, Prog. Theor. Phys. 66(4), 1169 (1981), H. Nishimori, Prog. Theor. Phys. 76, 305 1986

Lorenzo Dall'Amico et al Nishimori meets Bethe: a spectral method for node classification in sparse weighted graphs J. Stat. Mech. (2021) 093405

Hierarchical tree (Negative Curvature topology of Graph) of spin glass state

Def. 1 (Weighted non-backtracking matrix) Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and a function $f: \mathcal{E} \rightarrow \mathbb{R}$, so that $\forall e \in \mathcal{E}, f(e) = \omega_e$ is the weight corresponding to the edge e , the weighted non backtracking matrix $B \in \mathbb{R}^{2|\mathcal{E}| \times 2|\mathcal{E}|}$ is defined on the set of directed edges of \mathcal{G} as

$$B_{(ij),(k\ell)} = \delta_{jk}(1 - \delta_{i\ell})\omega_{k\ell}.$$

Local (hierarchical) Hessian

Def. 2 (Bethe-Hessian matrix) Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a function $f: \mathcal{E} \rightarrow \mathbb{R}$ so that $\forall e \in \mathcal{E}, f(e) = \omega_e$ and a parameter $x \in \mathbb{C} \setminus \{\pm\omega_{ij}\}_{(ij) \in \mathcal{E}}$, the Bethe-Hessian matrix $H(x) \in \mathbb{C}^{n \times n}$ is defined as

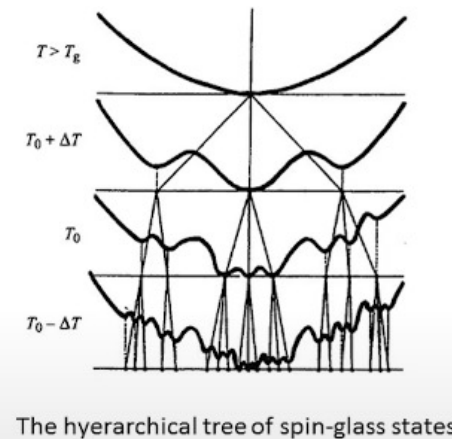
$$H_{ij}(x) = \left(1 + \sum_{k \in \partial i} \frac{\omega_{ik}^2}{x^2 - \omega_{ik}^2} \right) \delta_{ij} - \frac{x\omega_{ij}}{x^2 - \omega_{ij}^2}.$$

Since \mathcal{G} is an undirected graph, $H(x)$ is symmetric but not Hermitian, unless $x \in \mathbb{R}$. The relation between the spectra of the matrices B and $H(x)$ is given by the Watanabe-Fukumizu formula.

Prop. 1 (Watanabe-Fukumizu) Let $H(x)$ and B on the same graph \mathcal{G} and for the same weighting function f . $x \in \mathbb{C} \setminus \{\pm\omega_{ij}\}_{(ij) \in \mathcal{E}}$:

$$\det[xI_{2|\mathcal{E}|} - B] = \det[H(x)] \prod_{(ij) \in \mathcal{E}} (x^2 - \omega_{ij}^2),$$

x in the spectrum of B , $\det[H(x)] = 0$.



Nishimori Temperature estimation algorithm

To perform accurate node clustering, knowledge of the Nishimori temperature β_N is crucial for obtaining a precise estimate of the true node classes σ . A powerful estimator of σ is derived from the signs of the entries of the eigenvector \mathbf{x} of the Bethe-Hessian matrix $H_{\beta_N, \tilde{J}}$, associated with its smallest amplitude eigenvalue, which is close to zero.

Alg. 1: Nishimori Temperature estimation

Input: Weighted adjacency matrix of a graph $J \in \mathbb{R}^{n \times n}$, precision error $\varepsilon \in \mathbb{R}$

Return: Value of $\hat{\beta}_N \in \mathbb{R}^+$

- Compute c , the average degree of the underlying unweighted graph: $c = \frac{1}{n} \sum_i \sum_j \mathbb{I}(J_{ij} \neq 0)$
- Compute $\hat{\beta}_{SG}$ by solving $c \mathbb{E}[th^2(\hat{\beta}_{SG} J_{ij})] = 1$
- Set $t = 1$ and $\beta_t \leftarrow \hat{\beta}_{SG}$
- Initialize $\delta \leftarrow +\infty$

While $\delta > \varepsilon$:

- Compute $H_{\beta_t, J} = \delta_{ij} \left(1 + \sum_{k \in \partial i} \frac{th^2(\beta_t J_{ik})}{1 - th^2(\beta_t J_{ik})} \right) - \frac{th(\beta_t J_{ij})}{1 - th^2(\beta_t J_{ij})}$.
- Compute $\gamma_{min, t}$, the smallest eigenvalue of $H_{\beta_t, J}$, as well as its associated eigenvector \mathbf{x}_t
- Define the function $f_t(\beta') = \mathbf{x}_t^T H_{\beta', J} \mathbf{x}_t$, for $\beta' \in \mathbb{R}^+$
- Compute β_{t+1} by solving $f_t(\beta_{t+1}) = 0$
- Update $\delta \leftarrow |\gamma_{min, t}|$
- Increment $t \leftarrow t + 1$

return: β_{t-1}

Nishimori Temperature-Based Node Classification Algorithm

To perform accurate node clustering, knowledge of the Nishimori temperature β_N is crucial for obtaining a precise estimate of the true node classes σ . A powerful estimator of σ is derived from the signs of the entries of the eigenvector \mathbf{x} of the Bethe-Hessian matrix $H_{\beta_N, \tilde{J}}$, associated with its smallest amplitude eigenvalue, which is close to zero. First, consider \mathbf{y} , the eigenvector associated with the smallest eigenvalue of $H_{\beta_N, J}$. Let $A \in \{0,1\}^{n \times n}$ be the symmetric adjacency matrix of \mathcal{G} , defined by $A_{ij} = 1$ if $(ij) \in \mathcal{E}$, and $A_{ij} = 0$ otherwise. Let $D \in \mathbb{N}^{n \times n}$ be the diagonal degree matrix $D = \text{diag}(A\mathbf{1}_n)$.

$$\mathbb{E}[H_{\beta_N, J}] = I_n + \mathbb{E} \left[\frac{\text{th}(\beta J_{ij})}{1 - \text{th}^2(\beta J_{ij})} \right] (D - A).$$

Alg. 2 The Nishimori-Bethe relation for node classification

Input: Weighted adjacency matrix of a graph $\tilde{J} \in \mathbb{R}^{n \times n}$, precision error $\varepsilon \in \mathbb{R}$

Return: Value of $\hat{\beta}_N \in \mathbb{R}^+$, estimated label vector $\hat{\sigma} \in \{-1, 1\}^n$

- Shift the non-zero \tilde{J}_{ij} as: $\tilde{J}_{ij} \leftarrow \tilde{J}_{ij} - \frac{1}{2|\mathcal{E}|} \mathbf{1}_n^T \tilde{J} \mathbf{1}_n$
- Compute $\hat{\beta}_N \leftarrow \text{Compute_}\hat{\beta}_N$
- Compute $H_{\beta_t, J} = \delta_{ij} \left(1 + \sum_{k \in \partial i} \frac{\text{th}^2(\beta J_{ik})}{1 - \text{th}^2(\beta J_{ik})} \right) - \frac{\text{th}(\beta J_{ij})}{1 - \text{th}^2(\beta J_{ij})}$.
- Compute $\mathbf{x} \leftarrow$ the eigenvector associated to $\gamma_{\min}(H_{\hat{\beta}_N, \tilde{J}})$
- Estimate $\hat{\sigma}$ as the output of 2-class *k-means* on the entries of \mathbf{x}

return: $\beta_t, \hat{\sigma}$.

How to construct RBIM Model graph

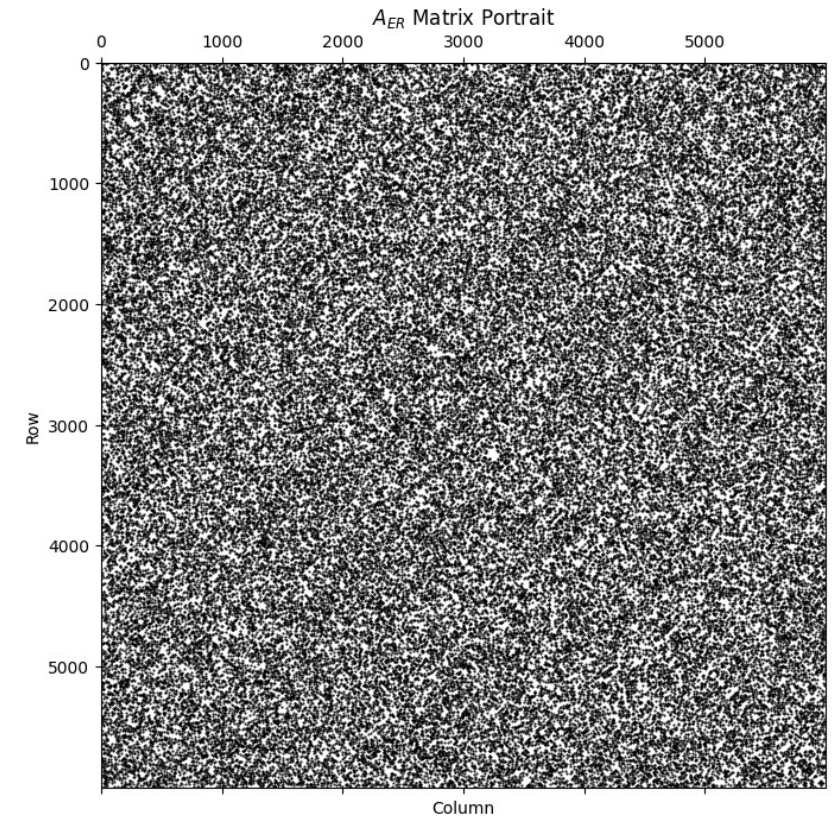
Baseline RBIM Erdos-Renyi (ER) graph

$G(n, p)$ has on average $\binom{n}{2} p$ -edges.

The distribution of the degree of any particular vertex is binomial (edge/bond degree):

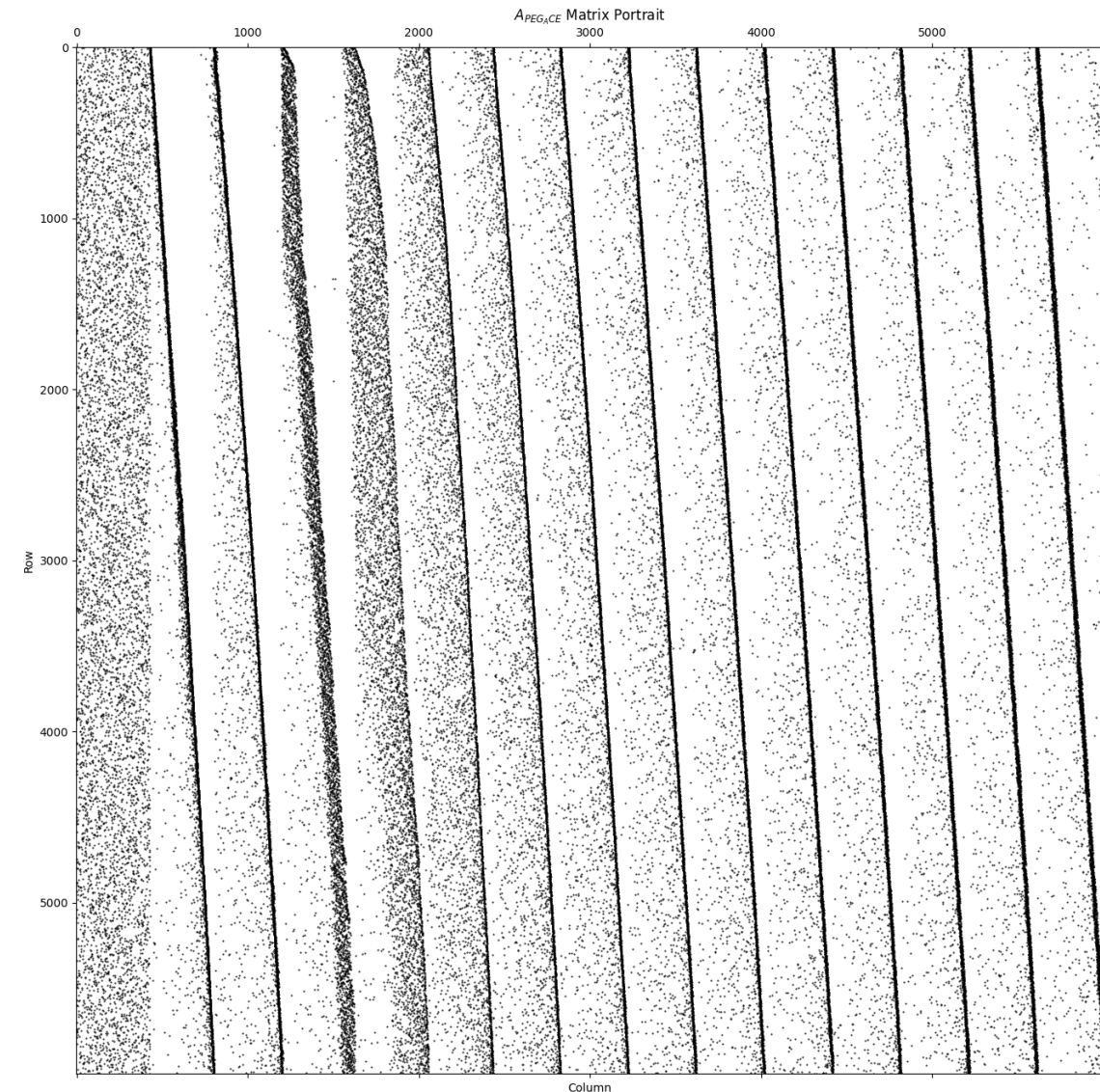
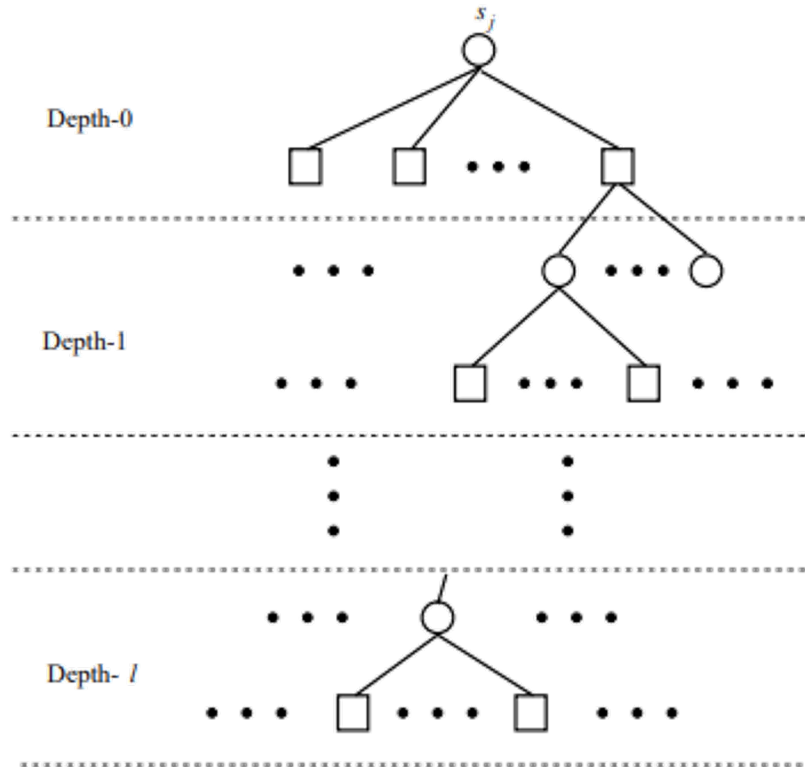
$$P(\deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k},$$

where n - the total number of edge in the graph.



Progressive-Edge Grown Algorithm with Approximate Message Degree (ACE) maximization for LDPC bipartite graph construction

Greedy choice of matrix position values for every variable nodes to increase girth of graph.



X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, Regular and irregular progressive edge-growth tanner graphs, IEEE Trans. Inf. Theory, vol. 51, no. 1, pp. 386-398, Jan. 2005.

Tian T., Jones C. R., Villasenor J. D. and Wesel R. D., "Selective avoidance of cycles in irregular LDPC code construction," in IEEE Trans. on Commun., vol. 52, no. 8, pp. 1242-1247, Aug. 2004.

Simulated Annealing method for Quasi-Cyclic LDPC bipartite graph construction

Algorithm 1 Simulated Annealing method for construction of QC-LDPC codes

Require: $M(\mathbf{H})$ —mother matrix, L —circulant size, g —girth of lifted matrix, EMD —minimal EMD value, I_{ter} — maximal number of iterations, $seed$ — a seed to be used in a pseudo-random number generator, $Temp$ — initial value of temperature

```

1:  $Nstep = 0$ 
2:  $i, j = rnd(seed)$ 
3: for  $it = 0; it \leq I_{ter}; it = it + 1$  do
4:   while  $M_{ij}(\mathbf{H}) = 0$  do
5:      $i, j = rnd(seed)$ 
6:   end while
7:   for  $k = 0; k \leq L - 1; k = k + 1$  do
8:      $\Theta_k = enumcircycles(i, j, k, g, EMD)$ 

```

$$P(k) = w(k) / \sum_{m=0}^{L-1} w(m),$$

$$w(k) = e^{\frac{-\Theta_k}{Temp}},$$

where Θ_k —number of cycles through $E_{ij}(\mathbf{H})$ -CPM with shift value k , $P(k)$ — probability of k -shift CPM value choice, $w(k)$ —probability weight function;

```

9:   end for
10:    $\Phi = enumcycles(E(\mathbf{H}), g, EMD),$ 

```

where Φ —total number of cycles in exponent matrix $E(\mathbf{H})$;

```

11:    $E_{ij}(\mathbf{H}) = rndshift(P, Temp)$ 

```

```

12:    $Nstep = Nstep + 1$ 

```

```

13:    $Temp = \eta \frac{\Phi}{Nstep^2},$ 

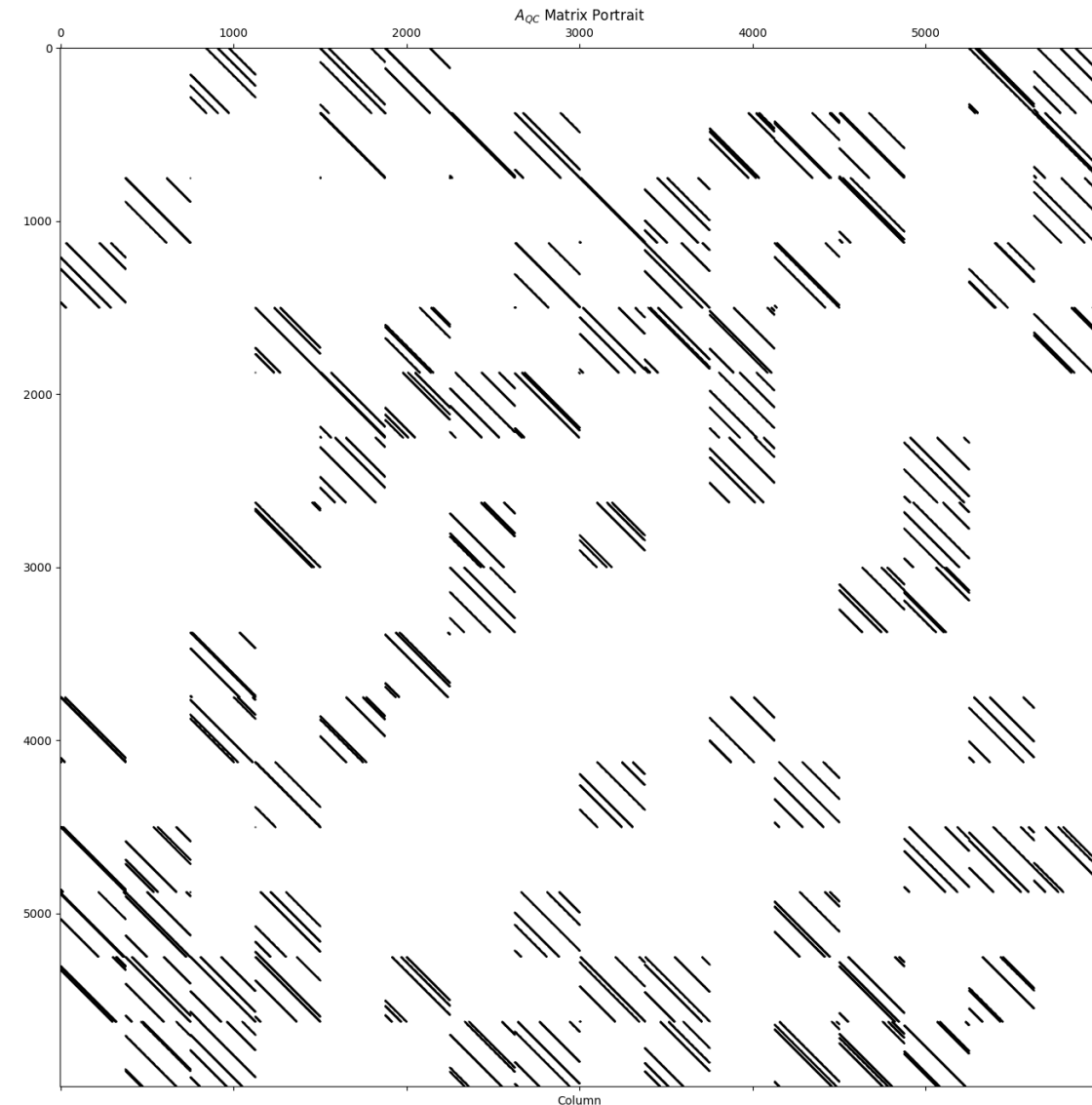
```

where η —some constant value;

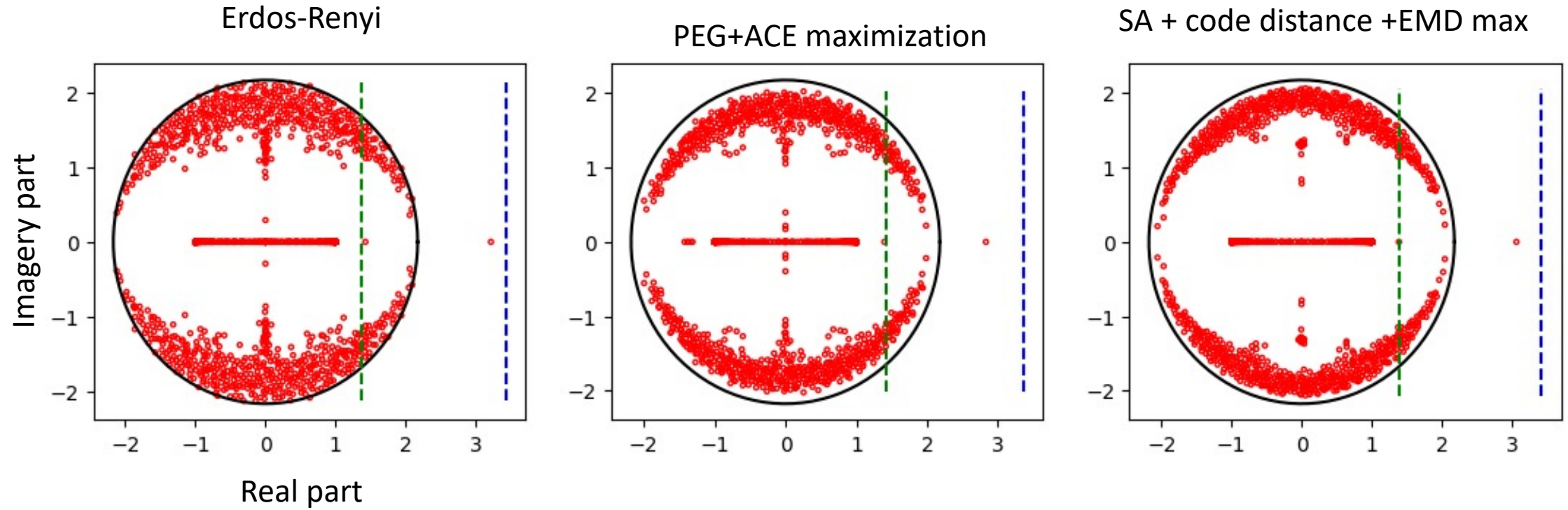
```

14: end for
    return  $E(\mathbf{H})$ 

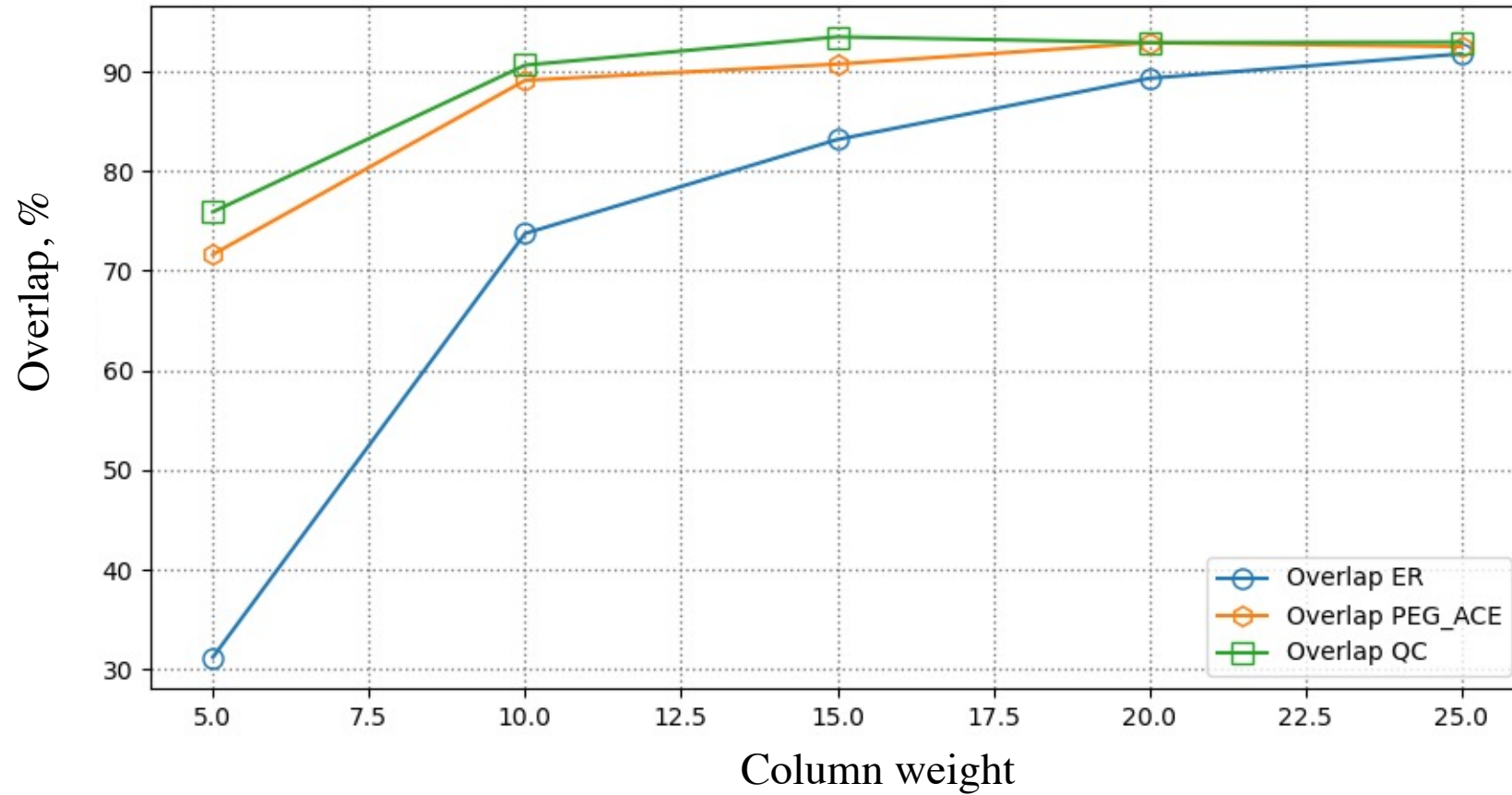
```



Spectrum of the weighed matrix B



Metric of unsupervised Nishimuri Temperature Clustering under RBIM



Application of the proposed RBIMS graphs for correction
generative adversarial attack on data

Non-target adversarial attack (data poisoning)

The goal of an untargeted adversarial attack (data poisoning) is to create a small disturbance in the input image that will cause the neural network to mistakenly assign it to some incorrect class.

Algorithm for non-targeted adversarial attack:

1. Compute the loss gradient with respect to the input data:

$$\mathbf{g} = \nabla_{\mathbf{x}} \ell(\mathbf{x}, y, \theta)$$

2. Calculate the disturbance by taking the sign of the gradient:

$$\boldsymbol{\eta} = \varepsilon \cdot \text{sign}(\mathbf{g})$$

3. Perform adversarial attack :

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \boldsymbol{\eta}$$

where $\mathbf{x} \in \mathbb{R}^{n \times m \times 3}$ - input RGB image size $n \times m$, $y \in \{1, 2, \dots, C\}$ - class label, $f(\mathbf{x}, \theta)$ - neural network with parameters θ , which outputs a vector of class probabilities, $\ell(\mathbf{x}, y, \theta)$ - a loss function that measures the discrepancy between predicted class probabilities and the true class label, ε - disturbance magnitude, a small positive scalar that controls the intensity of the disturbance,

$$\text{sign}(x) - \text{sign function, } \text{sign}(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} .$$

Example of non-target adversarial attack application

 \mathbf{x}

Класс: маргаритка

 η \mathbf{x}_{adv}

Класс: КОНСКИЙ КАШТАН



Original Image
Class: daisy



Perturbation



Adversarial Image (Epsilon = 1)
Class: buckeye



Example of non-target adversarial attack application

X
Класс: чайник

η

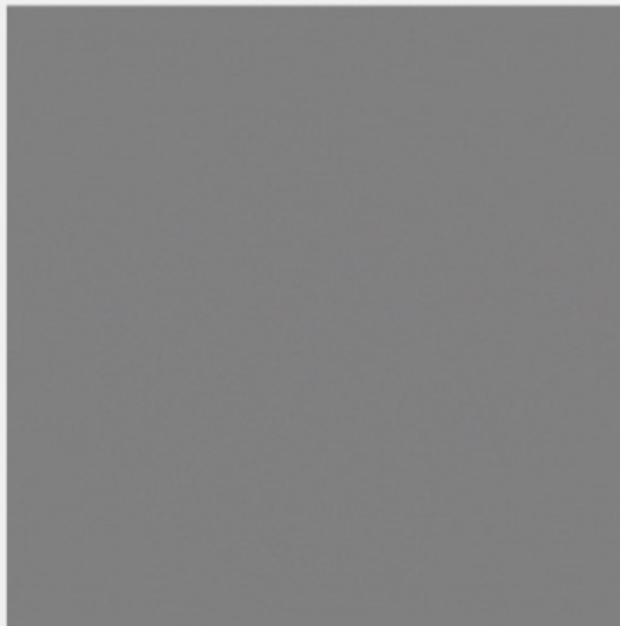
X_{adv}
Класс: кувшин

Original Image
Class: teapot



+

Perturbation



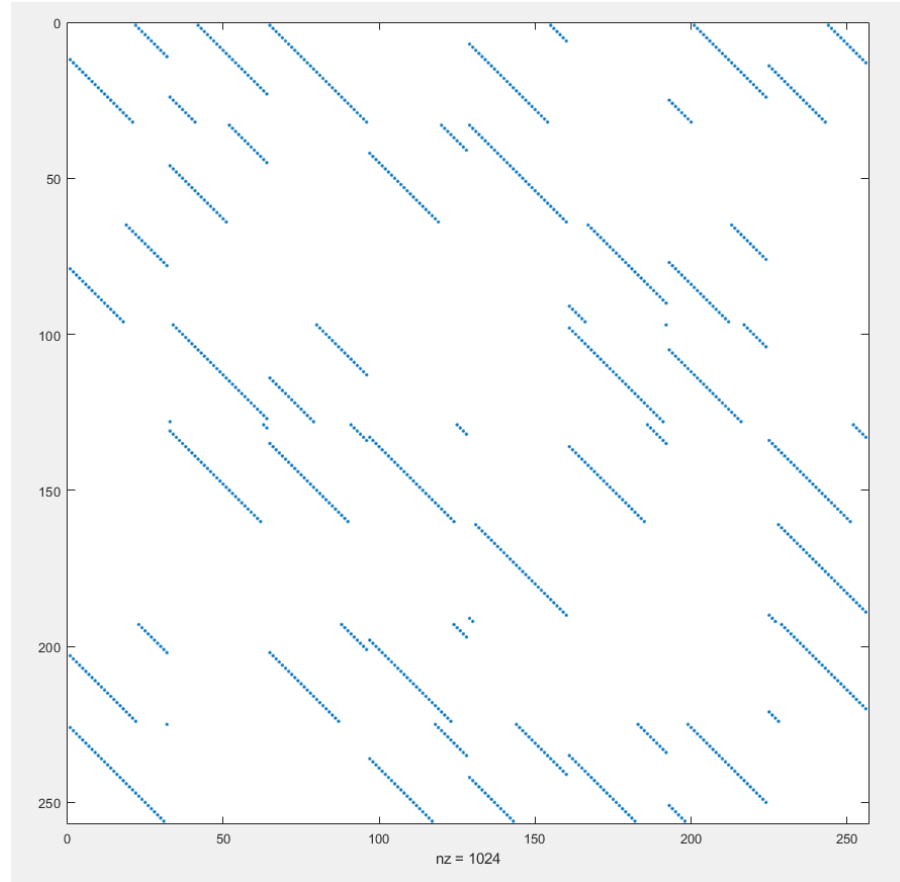
=

Adversarial Image (Epsilon = 1)
Class: pitcher

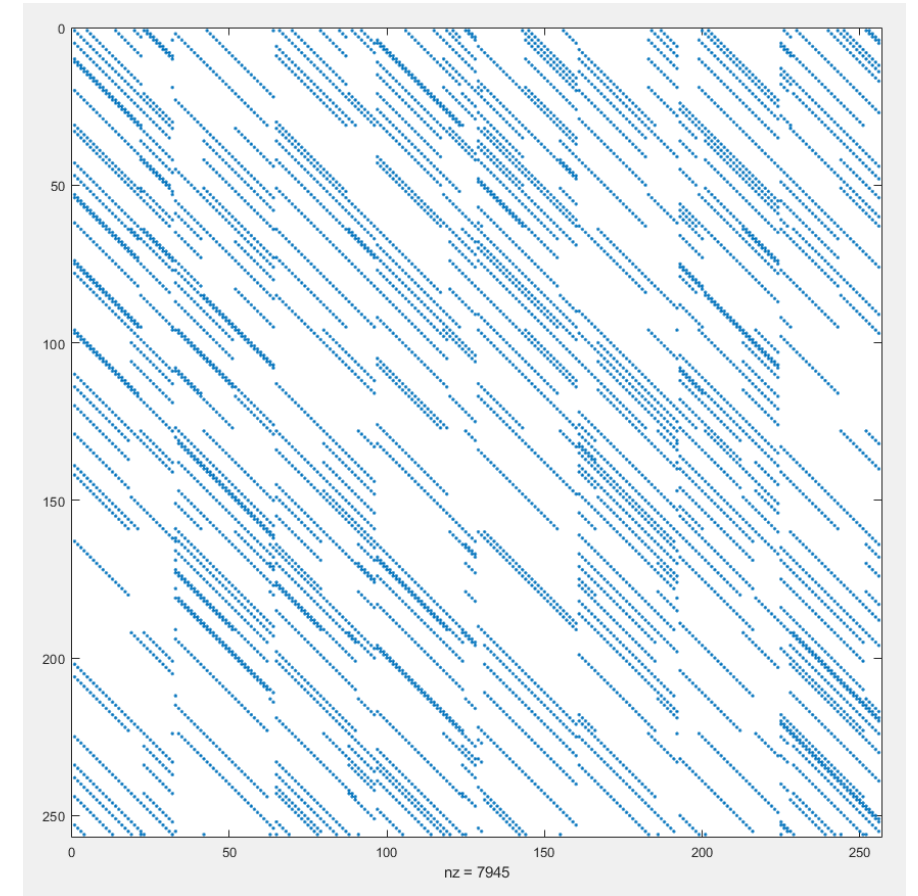


Application of the constructed mask of QC-LDPC code \mathbf{H} and its cyclic shift to \mathbf{x}_{adv}

$\mathbf{H}^{256 \times 256} =$



$\mathbf{H}_{shifted} =$



The \mathbf{H} construction method increases the code distance and EMD Spectrum*

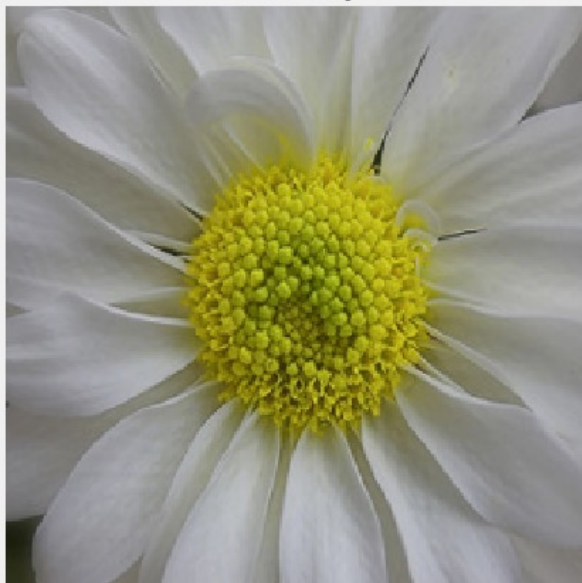
*:V. S. Usatyuk and S. I. Egorov, "Topology-Aware Sparse Factorization in Energy-Based Models: Tori QC-LDPC Codes and Circular Hyperboloid MET QC-LDPC Codes," *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, Moscow, Russian Federation, 2024, , pp. 1-6

Usatyuk V. S., Egorov S. I., "Heuristic Number Geometry Method for Finding Low-Weight Codeword in Linear Block Codes," *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, Moscow, Russian Federation, 2024, pp. 1-6

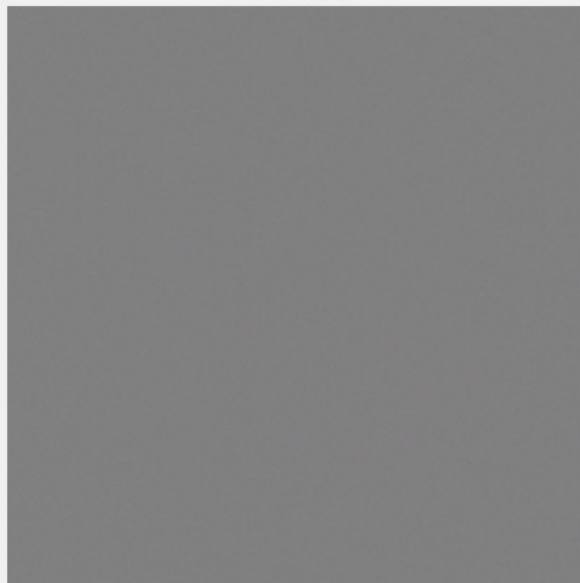
Application of the constructed mask of QC-LDPC code \mathbf{H} and its cyclic shift to \mathbf{x}_{adv}

 \mathbf{x}

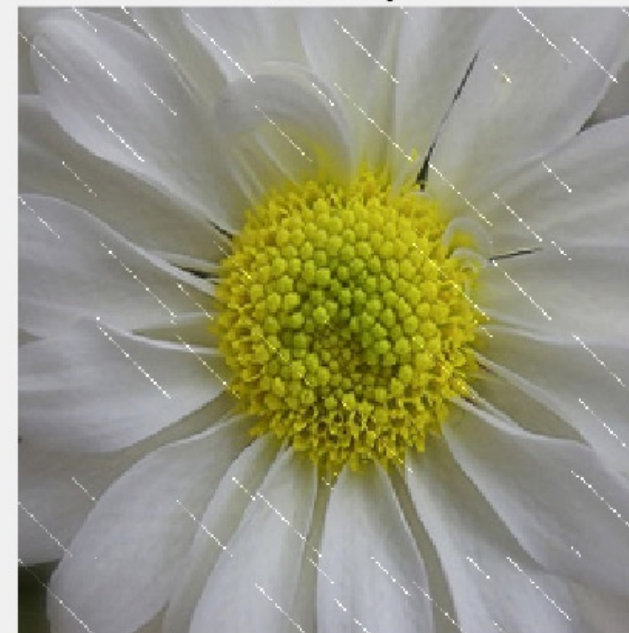
Класс: маргаритка

Original Image
Class: daisy η

Perturbation

 $\mathbf{x}_{adv_correct}$

Класс: маргаритка

Adversarial Image (Epsilon = 1)
Class: daisy

Application of the constructed mask of QC-LDPC code \mathbf{H} and its cyclic shift to \mathbf{x}_{adv}

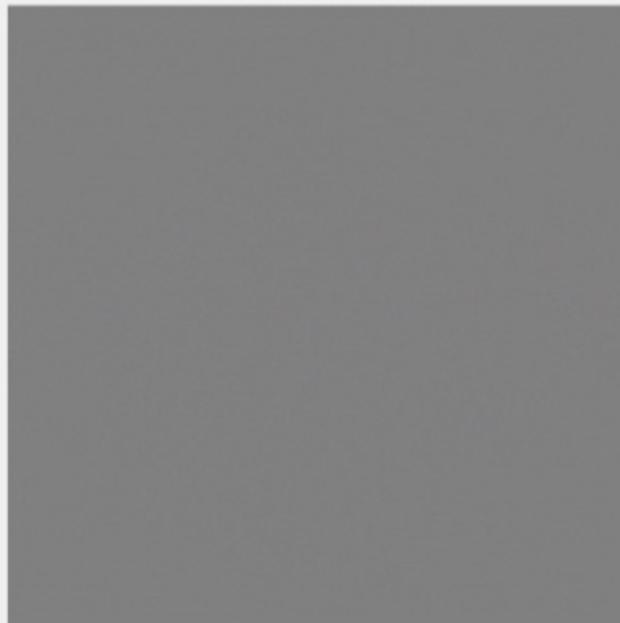
 \mathbf{x}

Класс: чайник

 η $\mathbf{x}_{adv_correct}$
Класс: чайникOriginal Image
Class: teapot

+

Perturbation

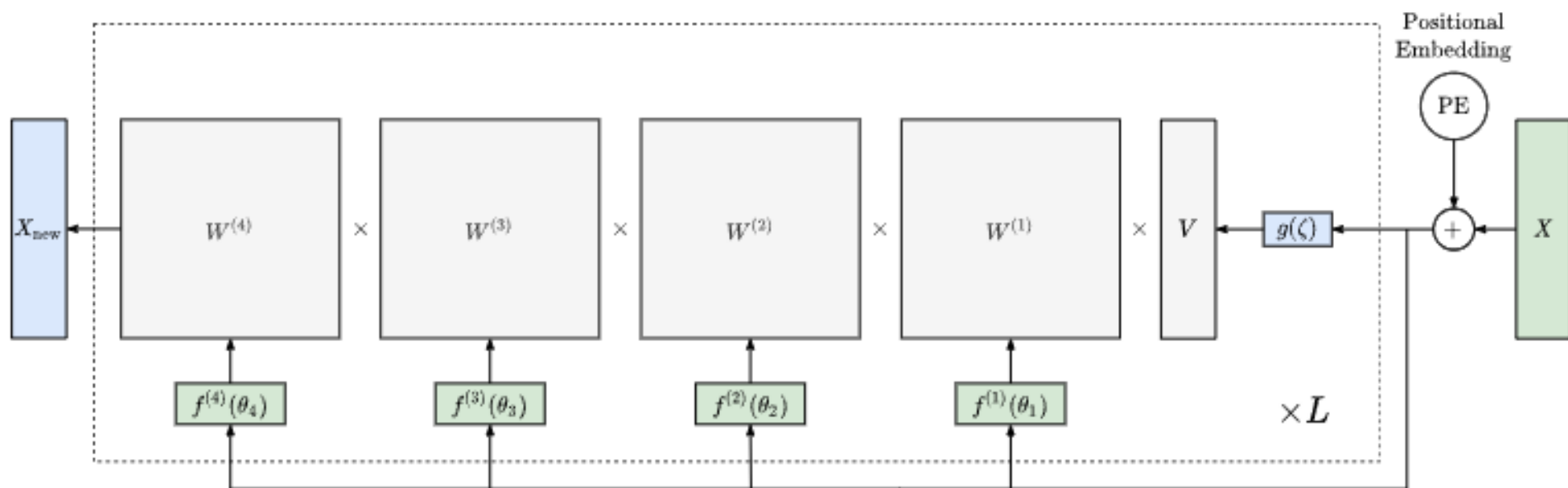


=

Adversarial Image (Epsilon = 1)
Class: teapot

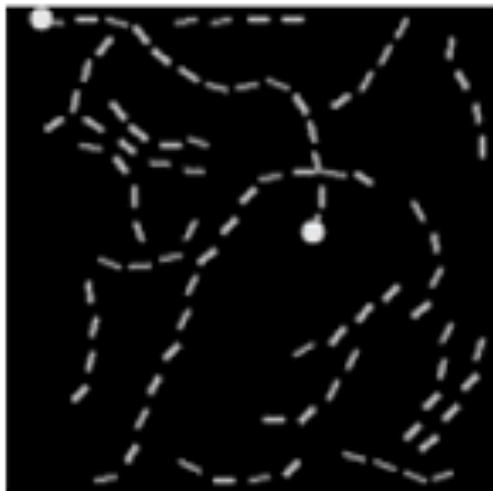
Application of the proposed RBIMs graphs for Transformer DNN

Example of Transformer DNN architectures

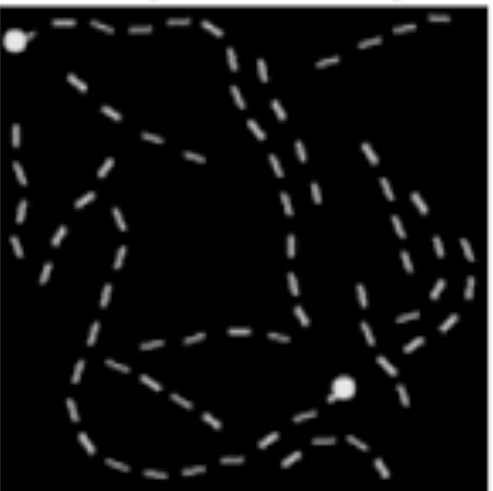


Complexity mostly depends from Attention matrix W

Competition Pathfinder from Long Range Arena



(a) A positive example.



1024 $q=256$ (gray) reshape to 1D vector.

Binary classification problem:

1 - you can connect two points inextricably along existing paths

0 - Can't connect the dots inextricably

Tay I. et al Long Range Arena: A Benchmark for Efficient Transformers

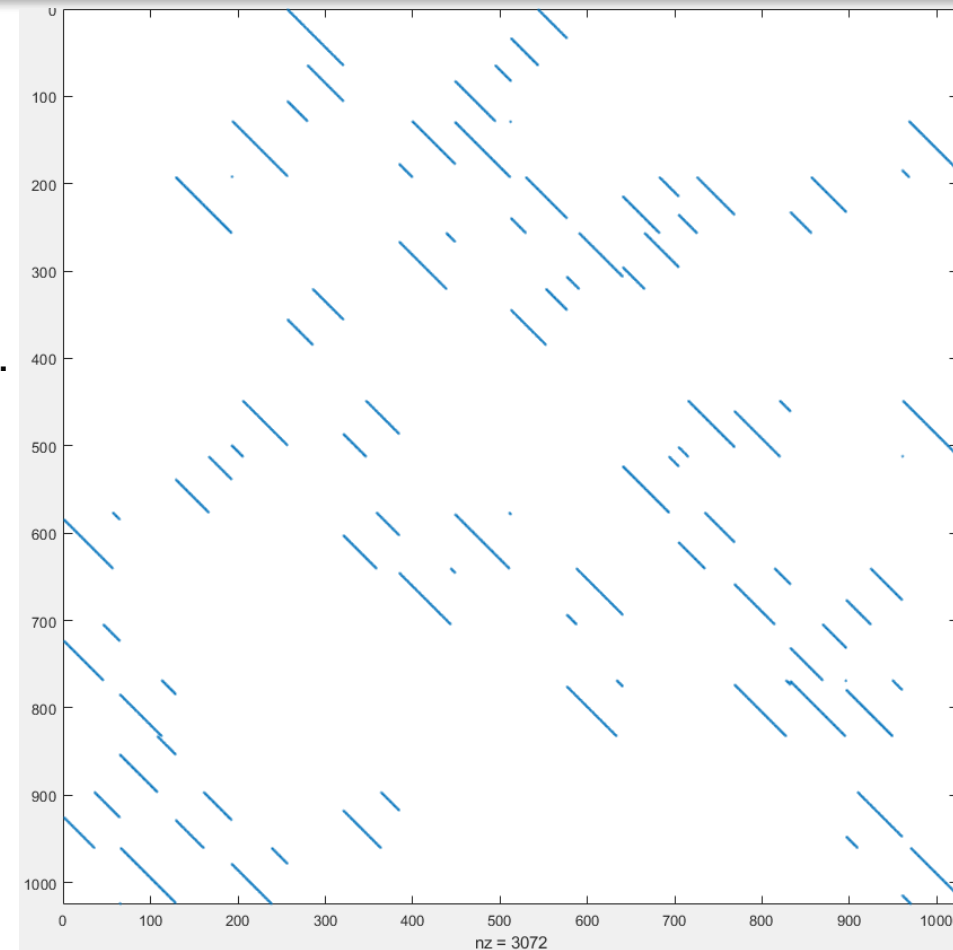
Attention matrix RBIM graph

Rank	Model	Pathfinder↑	Paper	Code	Result	Year	Tags
1	ChordMixer	96.67	ChordMixer: A Scalable Neural Attention Model for Sequences with Different Lengths			2023	
2	Mega	96.01	Mega: Moving Average Equipped Gated Attention			2022	
3	SGConv	95.46	What Makes Convolutional Models Great on Long Sequence Modeling?			2022	
4	S5	95.33	Simplified State Space Layers for Sequence Modeling			2022	
5	Liquid-S4	94.8	Liquid Structural State-Space Models			2022	
6	Mega-chunk	94.41	Mega: Moving Average Equipped Gated Attention			2022	
7	S4	94.20	How to Train Your HiPPO: State Space Models with Generalized Orthogonal Basis Projections			2022	
8	S4D	93.96	On the Parameterization and Initialization of Diagonal State Space Models			2022	

95.89% class. accur.

3d place at
Pathfinder

7 time less param
(RAM) and
40 less complexity



Attention matrix of size 1024 x 1024 contained 3072 non zero weights, Multiplexing using barrel shifter of size 64.
The trained network can be effectively used on FPGA VLSI and analog accelerators, as well as training.

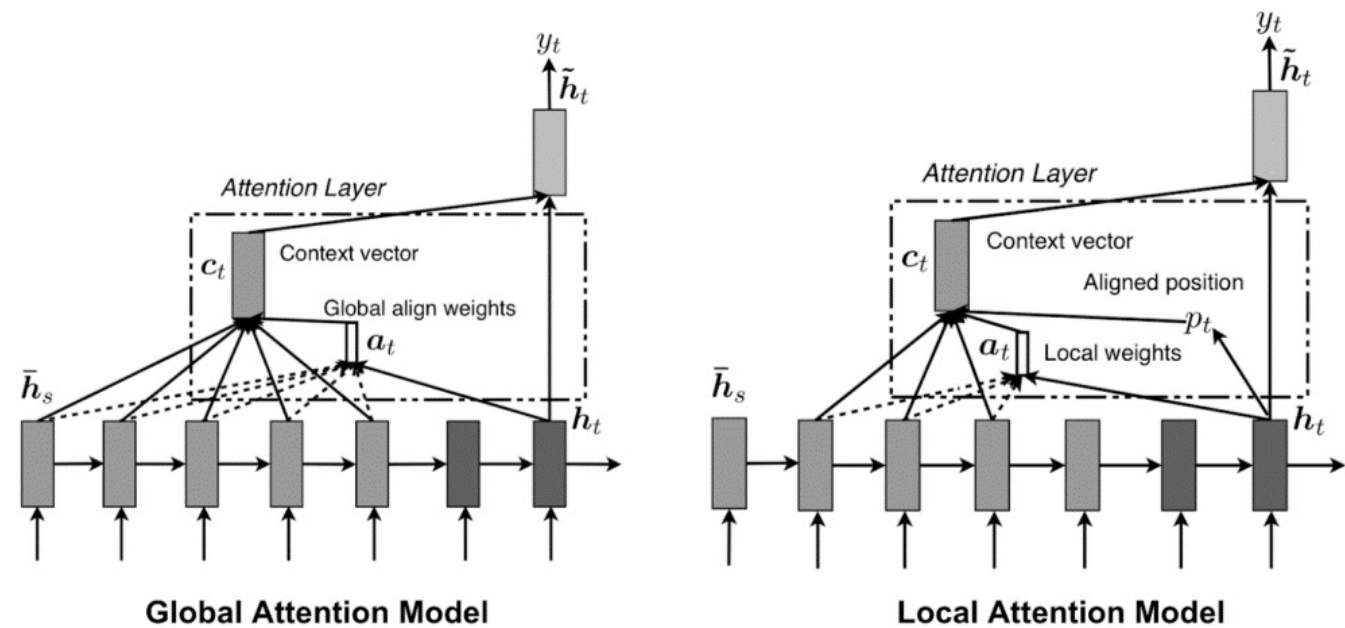
*<https://paperswithcode.com/sota/long-range-modeling-on-lra>

Summary

We proposed RBIM graph model construction methods which superior previous state-of-the art methods:

1. improve RBIMS clustering compare to Erdos-Renyi graphs, around 3 times less column weight and use Quasi-cyclic block structural matrix;
2. proposed one-shot defence from non-target adversarial attacks;
3. proposed simplification of visual transformer with 7 time less parameters (RAM) and around 40-times less complexity with lose 0.78% accuracy.

Thank You
For
Your Attention!



Cost of Nishimuri Temperature mismatch under binary classification
(image without noise and adversarial attack)
several percent, under unlimited iteration no more than 5% loss

Связь с нейронными сетями
Energy-Based Models and Neural Network

In an energy-based model, the probability of a configuration σ is defined in terms of an energy function, similar to the Hamiltonian of the Ising model. For the random bond Ising model, the probability $P(\sigma)$ of a spin configuration σ is given by the Boltzmann distribution:

$$P(\sigma) = \frac{e^{-\beta\mathcal{H}(\sigma)}}{Z},$$

where $\mathcal{H}(\sigma)$ is the Hamiltonian and Z is the partition function.

Similarly, deep neural networks can be used to model complex probability distributions. Consider a neural network with parameters θ that models a probability distribution $Q_\theta(\sigma)$. The goal is to approximate the true distribution $P(\sigma)$.

Variational inference involves approximating the true posterior distribution $P(\sigma)$ with a parameterized distribution $Q_\theta(\sigma)$ by minimizing the KL divergence between them. The KL divergence is defined as:

$$\text{KL}(Q_\theta \parallel P) = \sum_{\sigma} Q_\theta(\sigma) \log \frac{Q_\theta(\sigma)}{P(\sigma)}.$$

Substituting $P(\sigma)$ with its Boltzmann form, we get:

$$\text{KL}(Q_\theta \parallel P) = \sum_{\sigma} Q_\theta(\sigma) (\log Q_\theta(\sigma) + \beta\mathcal{H}(\sigma) + \log Z).$$

Since $\log Z$ is constant with respect to θ , it can be omitted from the optimization process. The optimization objective becomes:

$$\text{KL}(Q_\theta \parallel P) \propto \sum_{\sigma} Q_\theta(\sigma) (\log Q_\theta(\sigma) + \beta\mathcal{H}(\sigma)).$$

Training Neural Networks

The training of a neural network to approximate the distribution $P(\sigma)$ can thus be viewed as minimizing the KL divergence. This is equivalent to minimizing the variational free energy:

$$\mathcal{F}(\theta) = \sum_{\sigma} Q_{\theta}(\sigma) \left(\mathcal{H}(\sigma) + \frac{1}{\beta} \log Q_{\theta}(\sigma) \right).$$

Minimizing this free energy involves adjusting the neural network parameters θ to ensure that $Q_{\theta}(\sigma)$ closely approximates the true distribution $P(\sigma)$. This process is analogous to learning the weights in the neural network, where the energy function $\mathcal{H}(\sigma)$ can be interpreted as a **parameterized function (such as a neural network) that captures the dependencies between the variables.**

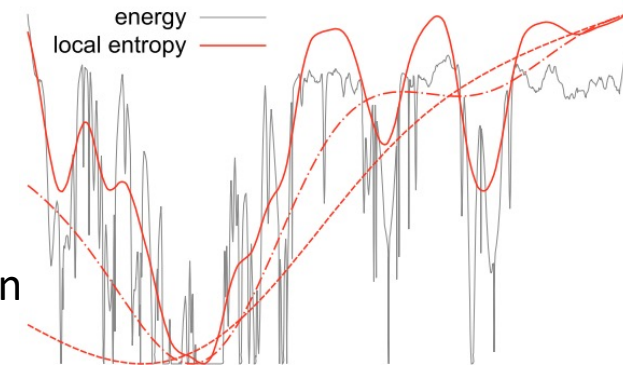
Local entropy*

$$\Phi(\sigma, \beta, \theta(\gamma)) = \log \sum_{\{\sigma'\}} e^{-\beta E(\sigma') - \gamma d(\sigma, \sigma')},$$

where $d(\cdot, \cdot)$ – distance between configuration, γ – weight, σ – reference spin configuration

Global extremum represented by symmetrical cycle in graph.

Local extremum around represented by odd degree connected cycle in graph.



Energy landscape compared with local entropy landscape

*Carlo Baldassia, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes *Proceedings of the National Academy of Sciences (PNAS)*, 113 (48), **November 15, 2016**, E7655-E7662

<https://www.pnas.org/doi/full/10.1073/pnas.1608103113>

Carlo Baldassia, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses. *Physical Review Letters* 115, 128101, p. 1-5., 2015

Usatyuk V., Sapozhnikov D., Egorov S. Spherical and Hyperbolic Toric Topology-Based Codes On Graph Embedding for Ising MRF Models: Classical and Quantum Topology Machine Learning <https://ui.adsabs.harvard.edu/abs/2023arXiv230715778U>

Quasi-Instatones– local minimum TS(a,b) under local entropy Φ solving an incorrectly posed operator problem*

$A \subset B, A$ – submatrices (local solution) contained in B connected through variable y_1, y_2, y_3 .

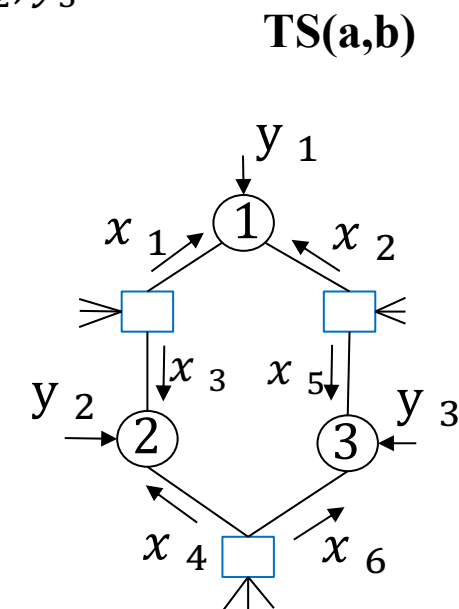
A subgraph with cycle 6

$x_1, x_2, x_3, x_4, x_5, x_6$

$$Ax = y,$$

$$\begin{cases} x_1 = (y_2 + x_4)/\alpha \\ x_2 = (y_3 + x_6)/\alpha \\ x_3 = (y_1 + x_2)/\alpha \\ x_4 = (y_3 + x_5)/\alpha \\ x_5 = (y_1 + x_1)/\alpha \\ x_6 = (y_2 + x_3)/\alpha \end{cases}$$

$$A = \begin{pmatrix} \alpha & 0 & 0 & -1 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & -1 \\ 0 & -1 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & -1 & 0 \\ -1 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & -1 & 0 & 0 & \alpha \end{pmatrix}$$



Choice of normalize α influences the convergence to global minima

For $\alpha > 1, A$ – has full rank and, accordingly, a unique solution

The selection of α allows you to find a solution to an ill-posed problem (the inverse of Tikhonov’s regularization**, “balances” the variables in the cycle with the subgraph external to them***)

VN ○

CN □

*Тихонов А. Н. О некорректных задачах линейной алгебры и устойчивом методе их решения // Доклады Академии наук СССР. — 1965. — Т. 163, № 3. — С. 591—594.

**М. И. Сумин, “О роли множителей Лагранжа и двойственности в некорректных задачах на условный экстремум. К 60-летию метода регуляризации Тихонова”, Вестник российских университетов. Математика, 28:144 (2023), 414–435

***M. Chertkov et al. Interaction screening: efficient and sample-optimal learning of ising models. NIPS’16 pp. 2603-2611

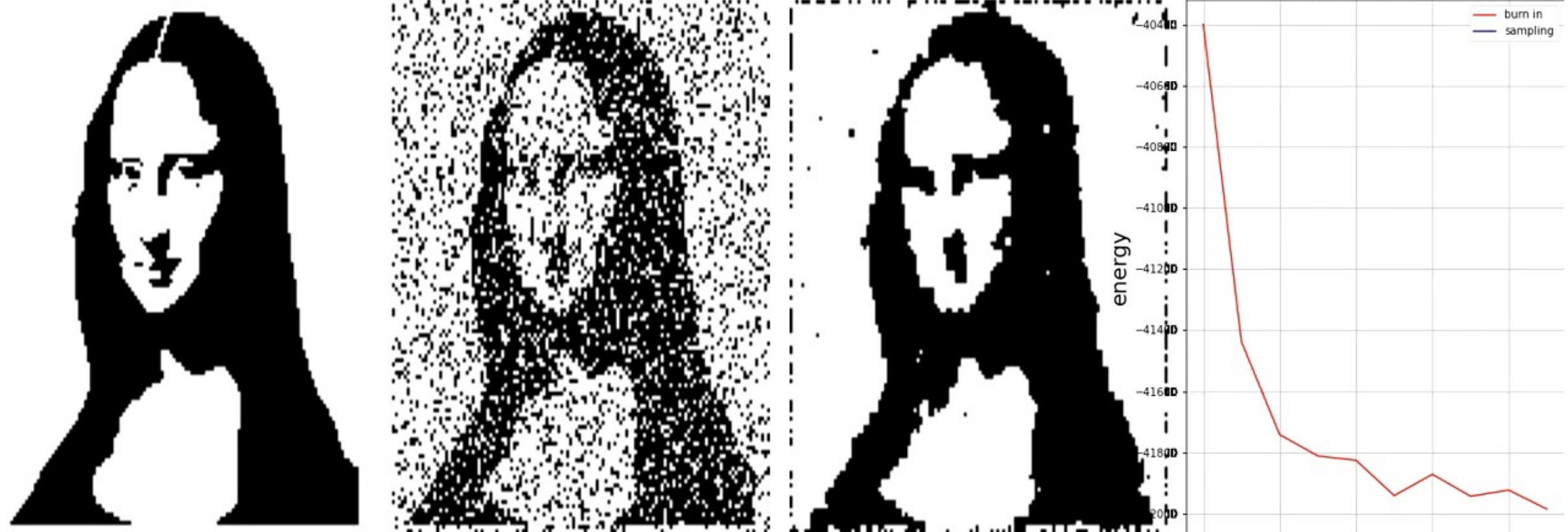
Denoising image with Gibbs Sampling, $\eta = 1$, $\beta = 1$, burn_in = 100, iter= 10

original

noisy input

denoised output (Gibbs)

energy (Boltzmann)



the minimum of $H_E = - \sum_{i=1, \dots, N} \sum_{a=1, \dots, M} C_{ij} J_{ij} \sigma_i \sigma_j$ under Nishimori Temp

The Weighted Laplacian Matrix (Laplacian Method)

A classical spectral clustering method for weighted graphs uses the *weighted Laplacian* matrix $L = \bar{D} - \tilde{J}$, where $\bar{D} = \text{diag}(|\tilde{J}| \mathbf{1}_n)$. The eigenvector associated with the smallest eigenvalue of L provides a relaxed solution to the NP-hard optimization problem of signed ratio-cut graph clustering.

For *signed graphs*, where \tilde{J} entries are ± 1 , the Bethe-Hessian matrix $H_{\beta, \tilde{J}}^{\text{signed}}$ is defined as:

$$H_{\beta, \tilde{J}}^{\text{signed}} = (1 - \text{th}^2(\beta))I_n + \text{th}^2(\beta)D - \text{th}(\beta)\tilde{J}.$$

As $\beta_N \rightarrow \infty$, $H_{\beta_N, \tilde{J}}^{\text{signed}} \rightarrow L$. The signed Laplacian can be seen as the zero temperature limit of the Bethe-Hessian matrix.

The Mean Field Approximation (Naïve Bayes)

Using a naive mean field approximation, the probability distribution is:

$$p_{\mathbf{m}}(\mathbf{s}) = \prod_{i \in \mathcal{V}} \frac{1+m_i s_i}{2},$$

where m_i is the average of s_i over the distribution.

The associated variational free energy is:

$$\tilde{F}_{\tilde{\mathbf{J}}, \beta}^{\text{MF}}(\mathbf{m}) = - \sum_{(ij) \in \mathcal{E}} \beta \tilde{J}_{ij} m_i m_j + \sum_{i \in \mathcal{V}} \sum_{s_i} \frac{1+m_i s_i}{2} \log \left(\frac{1+m_i s_i}{2} \right).$$

The Hessian of the free energy at the paramagnetic point $\mathbf{m} = \mathbf{0}$ is:

$$H_{\beta, \tilde{\mathbf{J}}}^{\text{MF}} = I_n - \beta \tilde{\mathbf{J}}.$$

Thus, the eigenvectors of $H_{\beta, \tilde{\mathbf{J}}}^{\text{MF}}$ are simply the eigenvectors of $\tilde{\mathbf{J}}$, making β irrelevant.

The Spin Glass Bethe-Hessian

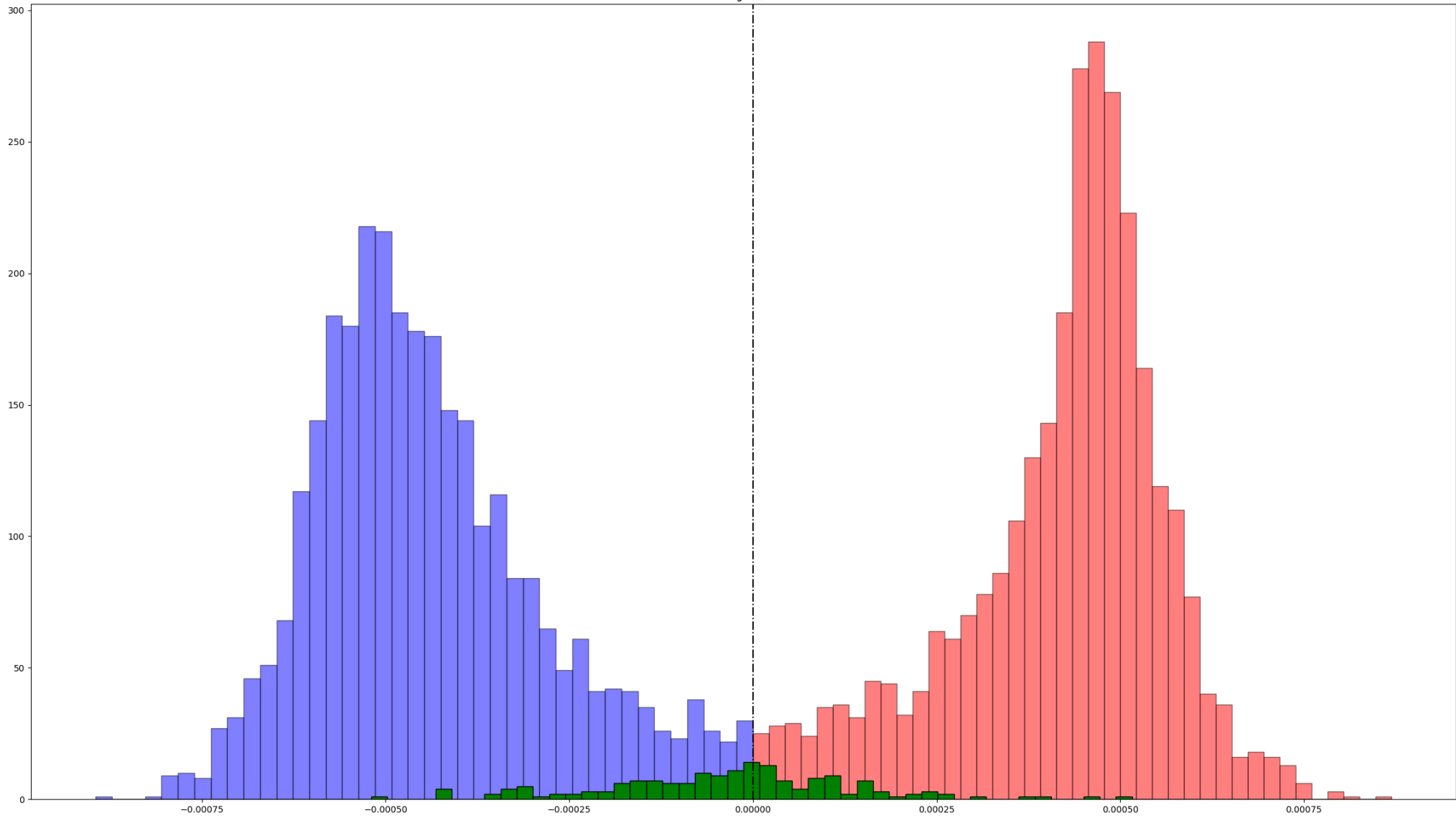
The spin glass Bethe-Hessian matrix, with $\beta = \beta_{SG}$, can also achieve non-trivial clustering as soon as theoretically possible. The temperature β_{SG} is estimated by solving:

$$c\mathbb{E}[\text{th}^2(\beta_{SG}\tilde{J}_{ij}\sigma_i\sigma_j)] = c\mathbb{E}[\text{th}^2(\beta_{SG}\tilde{J}_{ij})] = 1.$$

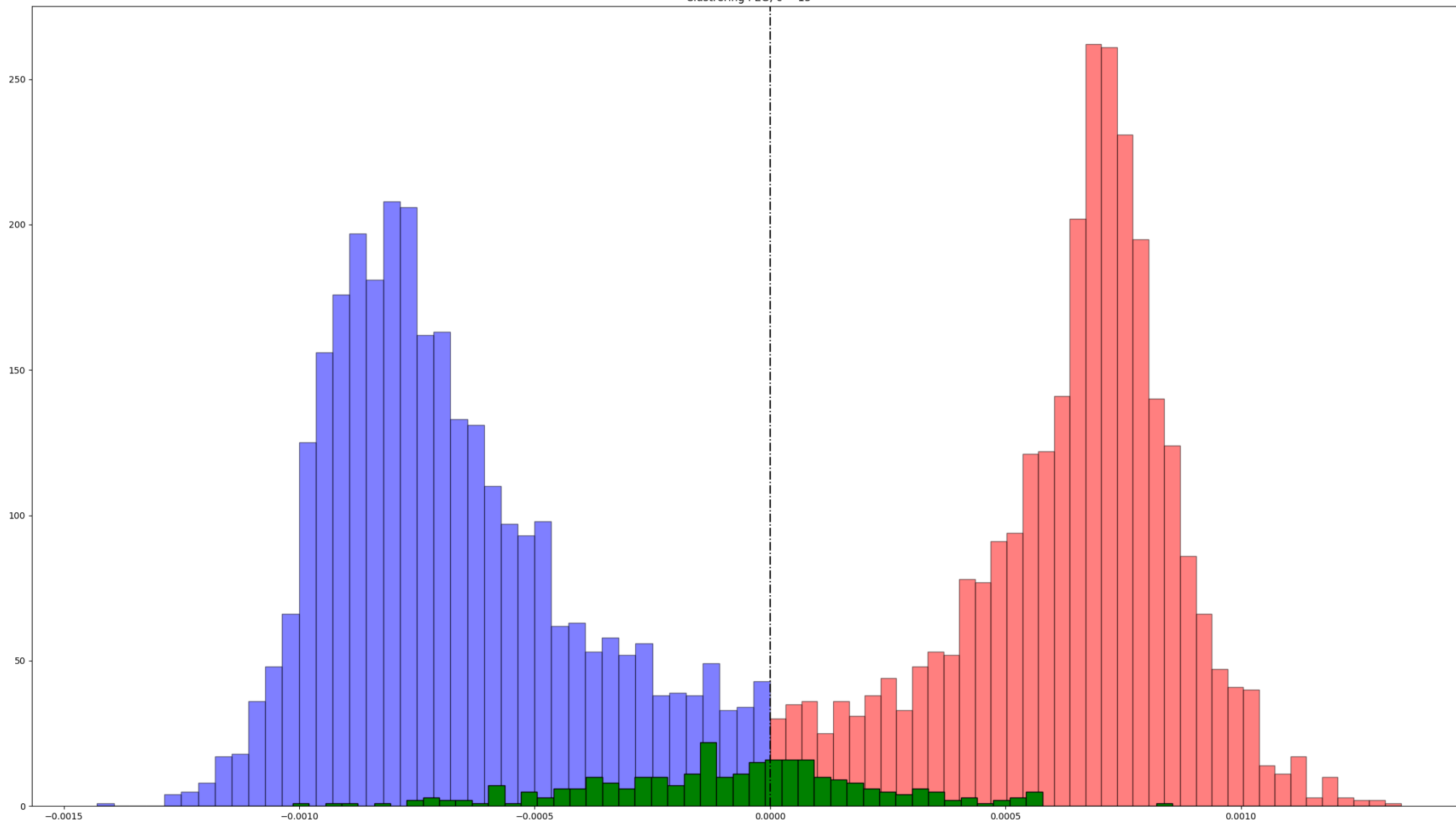
However, for realistic heterogeneous graphs, this choice may be suboptimal.

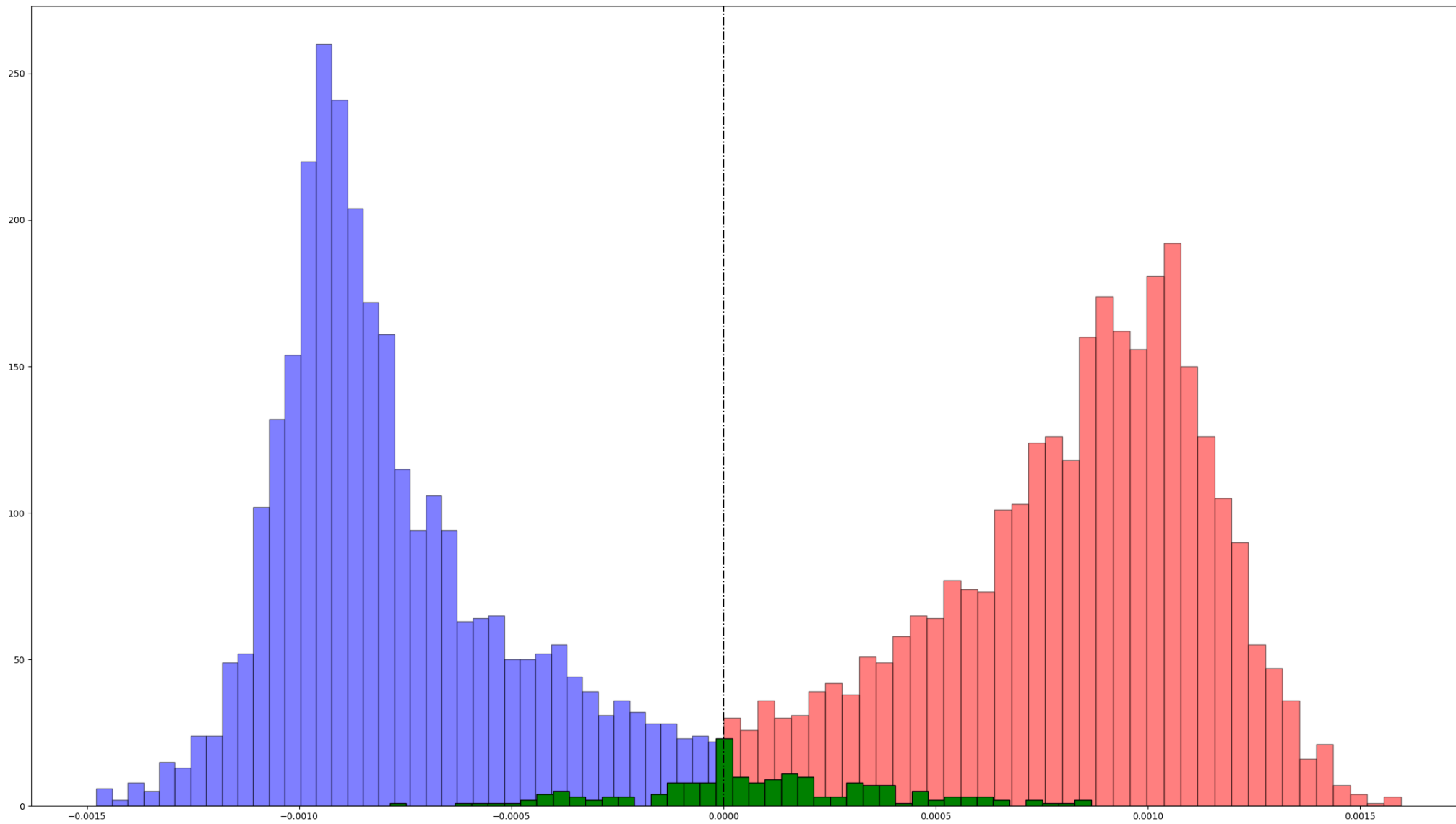
Matrix	Nishimuri	ISING	Mean Field Approx.	Laplace method
2_2_3000girth8upGirth6emd2seed212protograph_from_2x2.txt_matrix1.txt	23.67%	26.63%	0.03%	32.73%
2_2_3000girth8upGirth6emd2seed212protograph_from_2x2.txt_matrix2.txt	54.43%	47.00%	1.20%	0.03%
2_2_3000girth8upGirth6emd2seed212protograph_from_2x2.txt_matrix3.txt	53.40%	46.57%	1.10%	0.10%
2_2_3000girth8upGirth6emd2seed212protograph_from_2x2.txt_matrix4.txt	26.20%	29.10%	0.03%	31.97%
2_2_3000girth8upGirth6emd7seed212protograph_from_2x2.txt_matrix1.txt	23.67%	26.63%	0.03%	32.73%
2_2_3000girth8upGirth6emd7seed212protograph_from_2x2.txt_matrix2.txt	54.43%	47.00%	1.27%	0.03%
2_2_3000girth8upGirth6emd7seed212protograph_from_2x2.txt_matrix3.txt	53.40%	46.57%	1.10%	0.10%
2_2_3000girth8upGirth6emd7seed212protograph_from_2x2.txt_matrix4.txt	26.17%	29.10%	0.03%	31.97%
2_2_3000girth8upGirth8emd0seed212protograph_from_2x2.txt_matrix1.txt	23.67%	26.63%	0.03%	32.77%
2_2_3000girth8upGirth8emd0seed212protograph_from_2x2.txt_matrix2.txt	54.43%	47.00%	1.27%	0.03%
2_2_3000girth8upGirth8emd0seed212protograph_from_2x2.txt_matrix4.txt	26.17%	29.10%	0.03%	31.97%
4_4_1500girth6upGirth6emd6seed212protograph_from_4x4.txt_matrix1.txt	68.37%	67.70%	0.07%	0.03%
4_4_1500girth6upGirth6emd6seed212protograph_from_4x4.txt_matrix2.txt	69.70%	66.43%	0.30%	0.03%
4_4_1500girth6upGirth6emd6seed212protograph_from_4x4.txt_matrix3.txt	70.67%	67.83%	0.17%	67.63%
4_4_1500girth6upGirth6emd6seed212protograph_from_4x4.txt_matrix4.txt	68.60%	65.17%	0.17%	66.10%
4_4_1500girth6upGirth6emd7seed212protograph_from_4x4.txt_matrix1.txt	68.37%	67.70%	0.07%	0.03%
4_4_1500girth6upGirth6emd7seed212protograph_from_4x4.txt_matrix2.txt	69.70%	66.53%	0.30%	0.03%
4_4_1500girth6upGirth6emd7seed212protograph_from_4x4.txt_matrix3.txt	70.63%	67.77%	0.17%	67.63%
4_4_1500girth6upGirth6emd7seed212protograph_from_4x4.txt_matrix4.txt	68.60%	65.17%	0.17%	66.07%
4_4_1500girth8upGirth8emd0seed212protograph_from_4x4.txt_matrix2.txt	69.70%	66.20%	0.30%	0.03%
4_4_1500girth8upGirth8emd0seed212protograph_from_4x4.txt_matrix3.txt	70.67%	67.77%	0.17%	67.63%
4_4_1500girth8upGirth8emd0seed212protograph_from_4x4.txt_matrix4.txt	68.60%	65.17%	0.17%	66.07%
4_4_1500girth8upGirth8emd15seed212protograph_from_4_4_w2_1.txt_matrix2.txt	64.57%	63.57%	0.17%	0.03%
4_4_1500girth8upGirth8emd15seed212protograph_from_4_4_w2_1.txt_matrix3.txt	67.73%	66.40%	0.13%	0.03%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2.txt_matrix1(1).txt	90.60%	90.00%	0.20%	0.00%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2.txt_matrix1.txt	90.60%	90.00%	0.20%	0.00%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2.txt_matrix2.txt	90.57%	89.47%	0.23%	0.03%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2.txt_matrix3.txt	89.43%	89.03%	0.13%	0.03%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2_type2.txt_matrix1.txt	74.27%	62.33%	0.87%	0.03%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2_type2.txt_matrix2.txt	74.40%	62.90%	0.17%	0.03%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2_type3.txt_matrix1.txt	77.83%	65.80%	0.20%	0.03%
16_16_375girth6upGirth6emd0seed123protograph_from_16x16_5_w2_type3.txt_matrix2.txt	77.57%	65.57%	0.23%	0.03%
25_25_240girth6upGirth6emd6seed212protograph_from_25x25.txt_matrix1.txt	87.77%	85.80%	0.37%	87.03%
25_25_240girth6upGirth6emd6seed212protograph_from_25x25.txt_matrix2.txt	87.73%	86.67%	0.40%	0.03%
25_25_240girth6upGirth6emd7seed212protograph_from_25x25.txt_matrix1.txt	88.30%	86.23%	0.33%	87.20%
25_25_240girth6upGirth6emd7seed212protograph_from_25x25.txt_matrix2.txt	87.50%	85.73%	0.07%	0.03%
25_25_240girth6upGirth6emd8seed212protograph_from_25x25.txt_matrix1.txt	87.20%	85.70%	0.27%	85.87%
25_25_240girth6upGirth6emd8seed212protograph_from_25x25.txt_matrix2.txt	86.67%	86.43%	0.03%	0.03%
25_25_240girth6upGirth6emd9seed212protograph_from_25x25.txt_matrix1.txt	86.50%	85.20%	0.27%	85.00%
25_25_240girth6upGirth6emd9seed212protograph_from_25x25.txt_matrix2.txt	86.83%	86.07%	0.03%	0.03%
25_25_240girth6upGirth6emd10seed212protograph_from_25x25.txt_matrix1.txt	87.37%	85.97%	0.53%	86.63%
25_25_240girth6upGirth6emd12seed212protograph_from_25x25.txt_matrix13.txt	86.93%	85.57%	0.30%	86.13%
48_48_125girth6upGirth6emd12seed123protograph_from_48x48.txt_matrix1.txt	88.60%	87.73%	0.20%	87.23%

Clustering ER, c = 15



Clustering PEG, $c = 15$

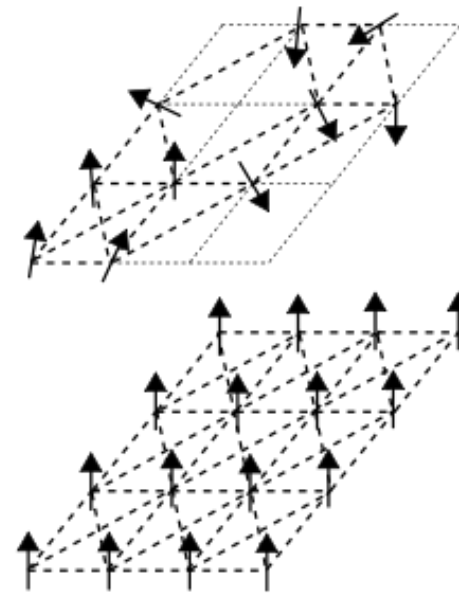




Двоичная 2x мерная модель спиновых стекол на графе $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$$\mathcal{H}(\sigma) = - \sum_{(i,j) \in \mathcal{E}} J_{ij} \sigma_i \sigma_j,$$

Each node $i \in \mathcal{V}$ is associated with a spin variable, $\sigma_i \in \{-1, +1\}$, \mathcal{E} - is the set of edges (bonds)
 J_{ij} - bond strengths, which are rand. assigned according to a specified probability distribution.
In the random bond Ising model, J_{ij} can take different values, typically following a distribution such as



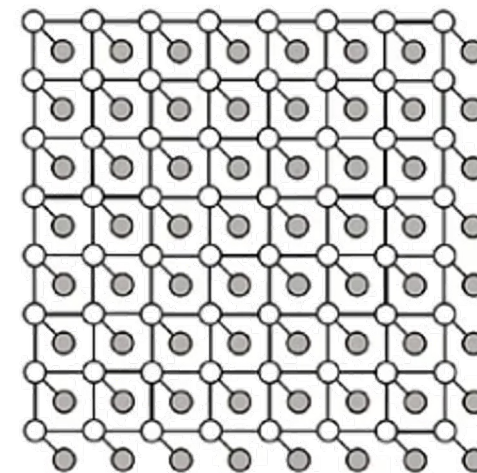
Задавая функцию, **можно закодировать значения исходного изображения**
отображения

$$\sigma_i = \exp(i\pi x_i) = \pm 1$$

$i = 1, \dots, N$, x_i - последовательность битов исходного изображения

Модель Изинга является примером марковского случайного поля (MRF) и возникла из статистической физики. Состояние каждого узла зависит от соседних узлов через потенциалы взаимодействия. В случае изображений это приводит к ограничению гладкости, т. е. пиксель предпочитает иметь тот же цвет, что и соседние пиксели.

Эта модель аналог черно-белого изображения



Модель Поттса (Potts q-ary 2D ISING)

Модель Домба-Поттса:

$$H_c = J_{ij} \sum_{i,j} \cos(\theta_{s_i} - \theta_{s_j}),$$

$\theta_k = \frac{2\pi k}{q}$ - матрица вращения, равномерно распределённая по углам, $s_j \in [1, \dots, q]$ - цвет частицы, J_{ij} - вес (потенциал) взаимодействия между спинами

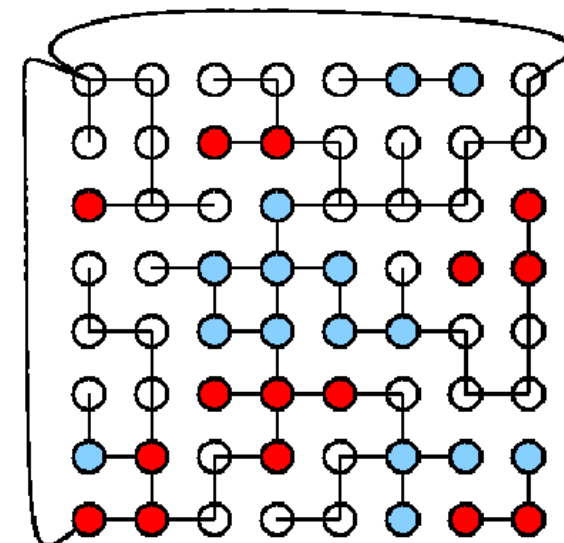
Модель Домба-Поттса состоит из спинов размещённых на геометрической решетке, вращение принимает q возможных значений равномерно распределённых под углами.

Стандартная модель Поттса:

$$H_p = -J_{ij} \sum_{i,j} \delta(s_i, s_j),$$

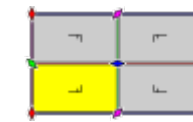
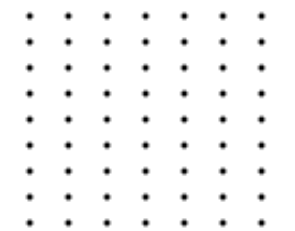
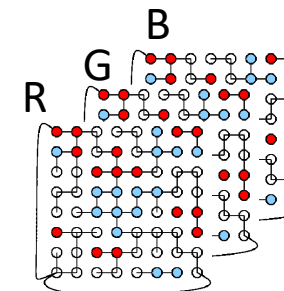
$\delta(s_i, s_j)$ - это функция Кронекера, равна 1, когда $s_i = s_j$, 0 в противном случае.

Аналог q изображения (q градации серого) и цветного изображения (5D=2D +3D для RGB)



Тернарная модель Поттса на квадратной решетке

[квадратная решетка](#)



Квазициклические Низкоплотностные коды и их ключевые свойства

$$H = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \begin{matrix} c_1 \\ c_2 \\ c_3 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

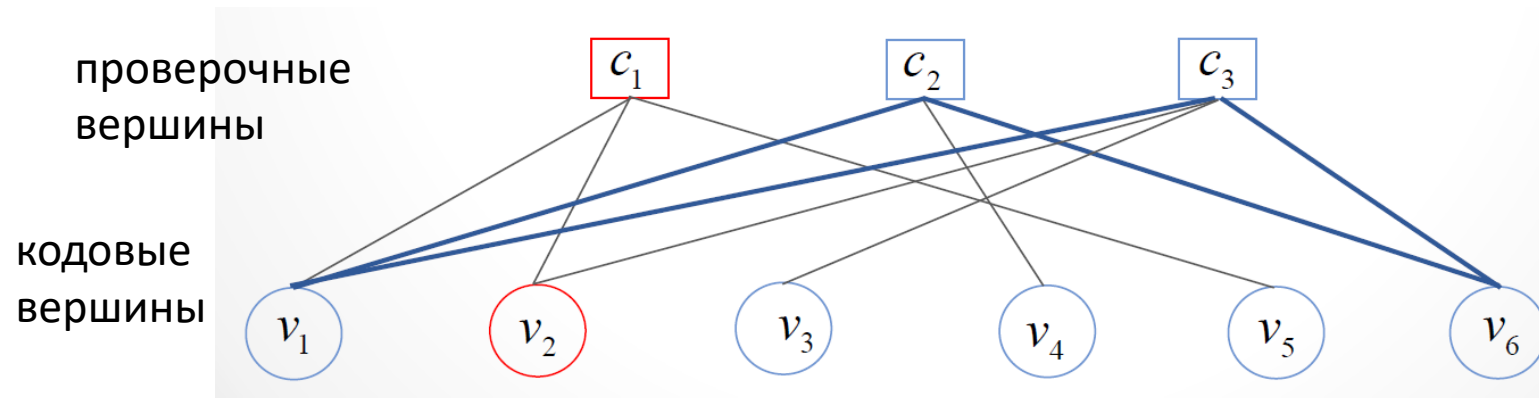
Низкоплотностные-коды (Low Density Parity Check, LDPC-коды), – это блочный линейный код размерностью k и длиной кодового слова n , задаваемый проверочной матрицей H размерностью $(n-k) \cdot n$, имеющей небольшую плотность отличных от нуля символов. Матрица размера $(n-k) \cdot n$ содержит порядка n проверок.

Каждая строка проверочной матрицы H задает уравнение проверки на четность:

$$\sum_{l=0}^{n-1} v_l \cdot h_{j,l} = 0$$

где j – номер строки проверочной матрицы (номер проверочного уравнения); l – номер символа кодового слова; $h_{j,l}$ – элемент проверочной матрицы.

Граф Таннера – это двудольный граф, соответствующей проверочной матрице LDPC- кода



Замкнутый простой путь в Таннер графе с узлами $v_1 \rightarrow c_2 \rightarrow v_6 \rightarrow c_3 \rightarrow v_1$ образуют цикл длины 4

Квазициклические низкоплотные коды (Quasi Cyclic-LDPC)

$$H = \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 1 \end{array} \begin{array}{l} c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \end{array}$$



$$H_{QC} = \begin{bmatrix} I^0 & I^1 & I^1 \\ I^0 & I^{-1} & I^0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

Квазициклический LDPC (коды QC-LDPC) - LDPC-коды с матрицей проверки на четность, определяемой структурированной блочной подматрицей - матрицей циклических перестановок.

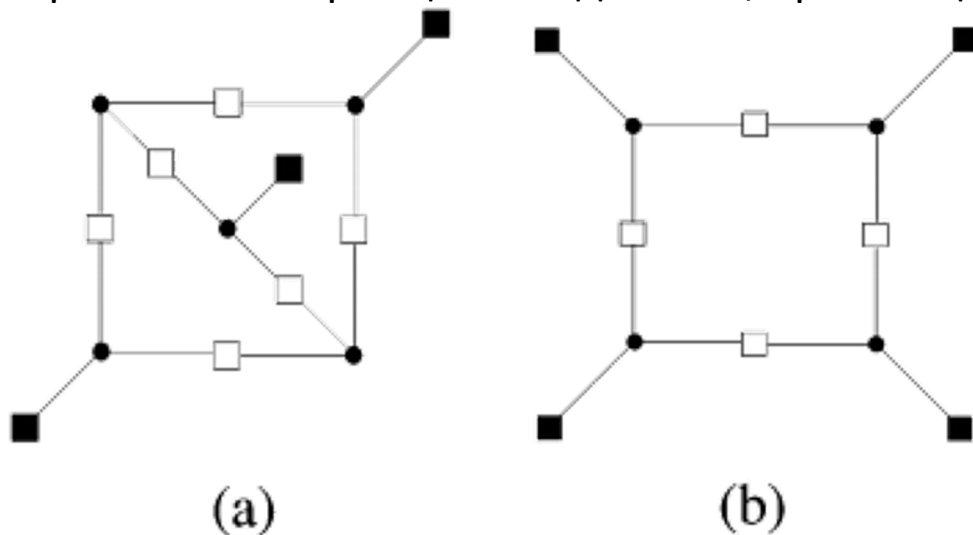
Матрица циклических перестановок размера 2x2, веса 1 и ее сжатая форма представления

$$I^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, I^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Традиционно используются матрицы веса 1.

Свойства низкоплотностных кодов. Трэппин-сети (a,b) , $a,b \in \mathbb{Z}$

На свойства низкоплотностных кодов отрицательно влияют циклы в графе Таннера, образующих «Трэппин-сети». (Trapping set, TS,) или (a,b) -подграфы (подграфы в графе Таннера, состоящие из a символьных узлов, b из которых инцидентны проверочным узлам с нечетными степенями). На рисунке ниже изображены два (a,b) -подграфа. Левый подграф образован пересечением трех циклов длины 8, правый одним циклом длины 8.



Trapping sets: (a) $(5, 3)$ and (b) $(4, 4)$

● -символьные узлы,

□ - проверочные узлы с четной степенью инцидентности

■ - проверочные узлы с нечетной степенью инцидентности

NP-сложная задача

A. McGregor and O. Milenkovic, "On the Hardness of Approximating Stopping and Trapping Sets in LDPC Codes," 2007 IEEE Information Theory Workshop, Tahoe City, CA, USA, 2007, pp. 248-253

Поиск Трэппин-сетов (a,b), b ≠ 0

*

Полный перебор требует анализ

Gurobi 7.5.1 solver линейного программирования 32 threads

$$\binom{4896}{48} \approx 8.2812e+115$$

SS 48 в коде Маргулиса (4896,2474) Margulis за 700 451 с.

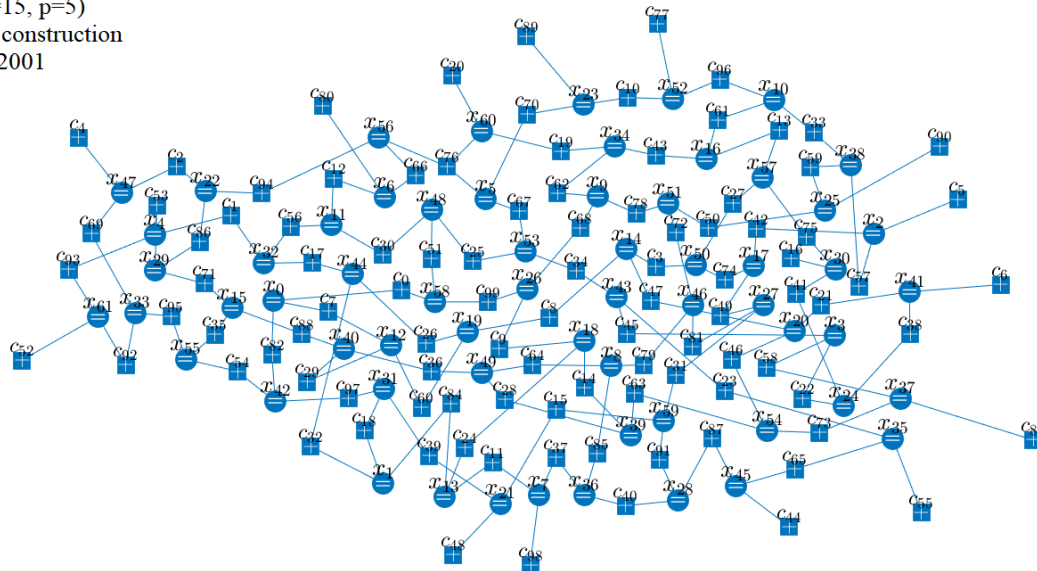
*Velasquez A., Subramani K., Wojciechowski P., On the complexity of and solutions to the minimum stopping and trapping set problems, Theoretical Computer Science, Volume 915, 2022, Pages 26-44,

Полный перебор требует анализ

$$\binom{4896}{62} \approx 1.2701E+143$$

Предложенный метод**

TS(62, 16) at Margulis(4896, 2448)
Ramanujan graph(q=15, p=5)
Vontobel-Rosenthal construction
Allerton 2000, ISIT 2001
true_K=2474



TS(62,16) with
TS variable nodes, (x_0)...(x_end) on Fig: 6 41 65 159 364 546
719 769 880 970 981 1097 1132 1140 1164 1274 1279 1385
1508 1561 1625 1681 1716 1819 1823 1979 1988 2194 2216
2369 2460 2506 2738 2765 2795 2855 2950 2976 3024 3154
3182 3271 3444 3566 3575 3836 3940 4012 4109 4141 4162
4355 4396 4401 4463 4547 4581 4671 4710 4800 4859 4864

CPLEX 12.8 LP solver 32 threads

Total (root+branch&cut) = 3.67 секунды. (292.92 ticks)

Кодовые слова - Трэппин-сети $TS(a,0)$, поиск слов малого(минимального веса)

Задача оценки кодового расстояния

Пусть матрица H , размера $(n - k) \times n$, заданная над F_q определяет линейное отображение $F_q^n \rightarrow F_q^{n-k}$ такое, что $x \rightarrow Hx$, $\ker(H) = \{c \in F_q^n \mid Hc = 0\}$.

Оценка кодового расстояния линейного блочного кода:
Найти вектор с минимальным число отличных от нуля координатных компонент, $c \in \ker(H) \setminus \{0\}$?

Пример поиска кодового расстояния - Трэппин-сети ($d_{min}, 0$)

$$Hc = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad GF(2)$$

$$\bar{c}_0^T = [0 \ 0 \ 0 \ 0 \ 0] \quad w(\bar{c}_0^T) = \|\bar{c}_0^T\| = 0$$

$$\bar{c}_1^T = [0 \ 0 \ 0 \ 1 \ 1] \quad w(\bar{c}_1^T) = \|\bar{c}_1^T\| = 2$$

$$\bar{c}_2^T = [0 \ 1 \ 1 \ 0 \ 1] \quad w(\bar{c}_2^T) = \|\bar{c}_2^T\| = 3$$

$$\bar{c}_3^T = [0 \ 1 \ 1 \ 1 \ 0] \quad w(\bar{c}_3^T) = \|\bar{c}_3^T\| = 3$$

Полином весового спектра кода

Слово веса 0

$$f(z) = 1 + z^2 + 2z^3$$

Весовой спектра кода

$$W(c) = [1, 0, 1, 2, 0, 0]$$

Слово веса 2

Слова веса 3

$$d_{min} = 2;$$

Спектр связности можно улучшать и в частности $TS(d_{min}, 0): d_{min} \rightarrow \max$

Международный конкурс по оценке кодового расстояния от объединения: Французского национального центра научных исследований (CNRS), Национального института исследований в области цифровых наук и технологий (Inria Paris), национального исследовательский институт математики и информатики в Нидерландах (CWI)

Best solutions

Weight	Authors	Algorithm	Details
214	Vasiliy Usatyuk	Lattice: Kannan emb, SBP (SBP), SVP	See details
215	Samuel Neves	-	See details
220	Valentin Vasseur	Dumer	See details

<https://decodingchallenge.org/low-weight/>

I. Dumer, D. Micciancio and M. Sudan, "Hardness of approximating the minimum distance of a linear code," in *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 22-37, Jan. 2003

Low-weight Codeword Problem

This page is dedicated to the problem of finding low-weight codewords for random binary linear codes.

Low Weight Codeword problem. Given integers n, k, w such that $k \leq n$ and $w \leq n$, an instance of the problem $LWC(n, k)$ consists of a full rank parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k) \times n}$. A solution to the problem is a non-zero vector \mathbf{c} with Hamming weight $\leq w$ such that $\mathbf{H}\mathbf{c}^T = \mathbf{0}$.

The challenge. Here, we focus on instances with code rate $R = 0.5$, that is $n = 2k$. We fix $n = 1280$ (see below). At this length, there exists on average a unique codeword of weight 144 (the [Gilbert-Varshamov bound](#)) and finding it should require at least 2^{128} operations (see below). Finding words of higher weight is easier. The goal is to find codewords with a weight as low as possible. The current record is Array.

Choice of n . The complexity of finding a codeword of weight equal to the Gilbert-Varshamov bound in a code of rate $R = 0.5$ using the [BJMM algorithm](#) asymptotically requires $2^{0.0999852n}$ operations, therefore with $n = 1280$, finding the smallest codeword should require at least 2^{128} operations.

Compared to the [Syndrome Decoding challenge](#), here the size of the instance is **cryptographically large**. The goal is to assess that finding codewords close to the [GV-bound](#) is hard. This problem is close to the [Shortest Vector Problem](#) for lattices.

Instance generation. The instances are generated using a [Python script](#). This script takes as input the length of the code and a seed (but for this challenge we only use the length $n = 1280$).

How to participate?

1. Choose an instance on the right. All instances have the same size, the only difference is the value of the random seed used to generate them. You can also use the [generator](#) to generate an instance with another random seed.
2. Parse the instance to get the values of \mathbf{H} . Find a non-zero codeword \mathbf{c} of weight $< \mathbf{Array}$ such that $\mathbf{H}\mathbf{c}^T = \mathbf{0}$.
3. Submit your solution using the [submission form](#). If your solution is correct, your name will appear in the hall of fame.

[Submit your solution](#)

[Hall of fame](#)

[Download instances](#)

[Instance generator](#)

[Format of instances](#)

[Download instances](#)

All instances have the same size. They are indexed by seed.

0	1	2	3
4	5	6	7
8	9	10	11

Best solutions

Weight	Authors	Algorithm	Details
211	Leo Ducas, Marc Stevens	Hybrid Wagner-Babai using Code Reduction [eprint:2020/869]	See details
212	Vasiliy Usatyuk	Lattice:SBP(BKZ), SVP	See details
214	Vasiliy Usatyuk	Lattice: Kannan emb, SBP (SBP), SVP	See details

NP-сложная задача

Модель Шеррингтона-Киркпатрика (Sherrington-Kirkpatrick)

$$H_E = - \sum_{i=1, \dots, N} \sum_{a=1, \dots, M} C_{ij} J_{ij} \sigma_i \sigma_j$$

C_{ij} - матрица связности, элемент которого равен 1, если два спина взаимодействуют, и 0 в противном случае,
 J_{ij} - вес (потенциал) взаимодействия между спинами, σ_i - спины модели Изинга

H_E можно преобразовать:

$$H_{SK} = - \sum_p \sum_{i_1, \dots, i_p=1, \dots, n} C_{i_1, \dots, i_p}^{(p)} J_{i_1, \dots, i_p} \sigma_{i_1} \dots \sigma_{i_p},$$

где $J_0^{(p)}$, $\Delta J_{(p)}^2$ М.О. и дисперсия J_{ij} .

Проверочная матрица квазициклического кода H определяет $C_{i_1, \dots, i_p}^{(p)}$.

x_i n -битное кодовое слов кодирующее спин $\sigma_i = 2x_i - 1$.

Взаимодействие пар спинов (без кратных ребер) происходит на глубину p .

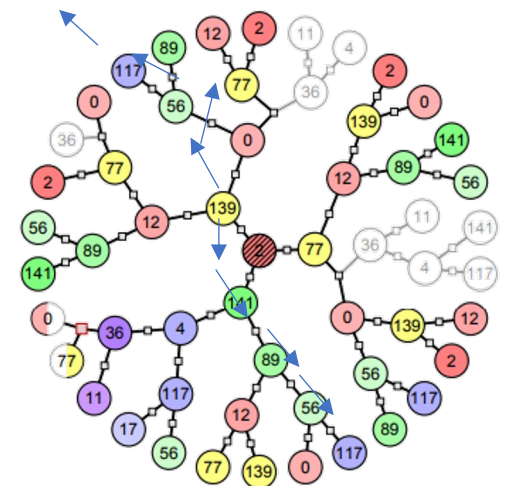
J_{i_1, \dots, i_p} - двух спинового взаимодействия и принимаются как независимые

случайные величины с известным распределением вероятностей.

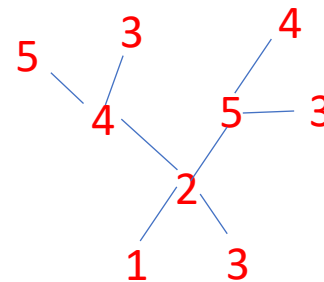
$$\bar{x}_0^T = [0 \ 0 \ 0 \ 0 \ 0], \bar{x}_1^T = [0 \ 0 \ 0 \ 1 \ 1]$$

$$\bar{x}_2^T = [0 \ 1 \ 1 \ 0 \ 1], \bar{x}_3^T = [0 \ 1 \ 1 \ 1 \ 0]$$

Множество спинов соответствующих кодовым словам является состояниями минимальной энергии Гамильтониана



Число слоев p — это число Пар спинов соединённых Проверочной матрицей кода. Число итераций декодирования



$$Hx = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Граф Таннер (мультиграф, в случае наличия не тривиального X_1)

Квазициклического низкоплотностного кода соответствует калибровочному поля

Кодовые вершины Ребра Графа Проверочные узлы

$$X = (X_0, X_1, X_2)$$

$$X_1 \subset X_0 \times X_2$$

$$(j, \alpha) \in X_1 \Leftrightarrow j \in \alpha \quad \alpha = 1, \dots, M$$

$$|X_0| = N \quad |X_2| = M$$

$$|X_1| = \sum_{j \in X_0} m_j = \sum_{\alpha \in X_2} n_\alpha$$

Кодовые слов (закодированные в спин)

$$\sigma = \{\sigma_j\}_{j \in X_0}; \sigma_j = \pm 1$$

Эквивалентные коды (калибровочная инвариантность)

$$X \sim X' \Leftrightarrow C_X = C_{X'}$$

$$C_X = \left\{ \sigma \left| \prod_{j \in \alpha} \sigma_j = 1, \forall \alpha \in X_2 \right. \right\}$$

Калибровочная группа

$$G = GL(M, F_2)$$

Эквивалентные коды (калибровочная инвариантность)

$$Hx = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H'x = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Графы не эквивалентны, мы добавили цикл

Статистическая модель независимых ошибок спина

Намагниченности спинового стекла

Вероятность результата измерения

$$P_j(\zeta) = (2\pi s^2)^{-N/2} \int dh \delta(m_j(h) - \zeta) \exp(-F(h))$$

Вероятность индивидуальных ошибок спина

Для симметрично распределенного Гауссова шума

$$B_j = \int_{-\infty}^0 d\zeta P_j(\zeta)$$

$$F(h) = (2s^2)^{-1} \sum_{j \in X_0} (h_j - s^2)^2$$

Отношения сигнал-шум (ОСШ), Температура

В функции плотности вероятности преобладает наиболее вероятная конфигурация шума инстантон (седловая точка)

$$\left. \frac{\partial}{\partial h} (F(h) - \lambda(m_j(h) - \zeta)) \right|_{h=\bar{h}} = 0$$

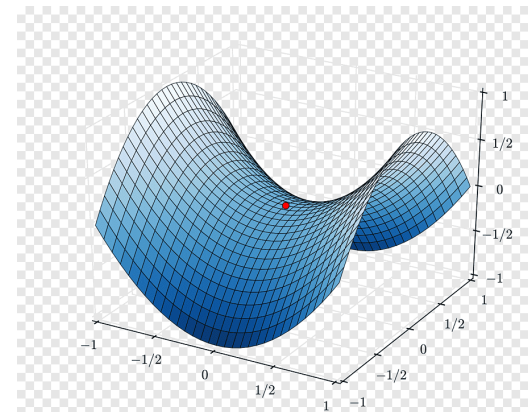
$$m_j(\bar{h}) - \zeta = 0$$

Множитель Лагранжа

$$P_j(\zeta) \approx \kappa^{-1}(\zeta) \exp(-F(\bar{h}))$$

$$B_j \approx \kappa_0^{-1} \exp(-F(\bar{h}))$$

TS(a,0)
Кодовые слова



Топологические инварианты и кривизна Изинг модели полученной из графов Таннера (мультиграфа в случае веса циркулянта больше 1)

Локальная система:

Каждый узел имеет древовидную "окрестность"

$$j \in U_j^{(l)} \subset X$$

Фундаментальная группа

свободная группа с генератором

$$\bar{X}(a; l) \subset \bar{X}; a \in p^{-1}(j) \cong \pi_1(X) \cong F(g)$$

Теорема Черна-Гаусса-Бонне

$$\sum_{j \in X_0} \frac{2 - m_j}{2} + \sum_{\alpha \in X_2} \frac{2 - n_\alpha}{2} = 1 - g$$

Локальная кривизна

Род поверхности (genus)



Род поверхности 3 (genus)

Графы с постоянной кривизной

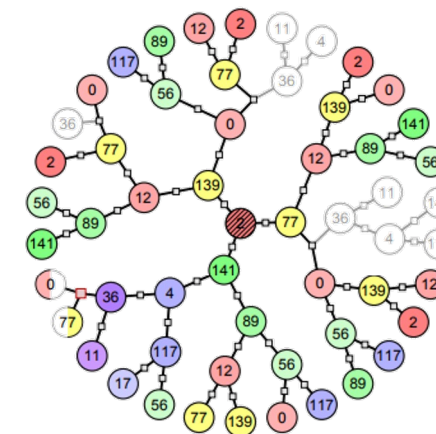
$$m_j = m \quad mN = nM$$

$$n_\alpha = n \quad \bar{X} \cong Y(m, n)$$

Универсальное покрывающее дерево

(аналогично римановым поверхностям)

$$p: \bar{X} \rightarrow X$$



Покрывающее дерево универсально и обладает высокой симметрией

Блочной структурой

Wiberg Codes and iterative decoding on general graphs 1995

Weiss Correctness of Local Probability Propagation in Graphical Models with Loops 2000

Knill A GRAPH THEORETICAL GAUSS-BONNET-CHERN THEOREM arXiv:1111.5395 2011

Forney, "Codes on Graphs: Models for Elementary Algebraic Topology and Statistical Physics 2018

Meshulam, Roy. (2018). Graph codes and local systems.

Квази-инстантоны – псевдокодированные слова TS(a,b)

Дискретные значения спина

$$\sigma_j = \prod_{i \in \gamma_j}^{i \neq j} \text{sgn}(\bar{\eta}_i); j \in A_0(\bar{\eta}) \quad A(\bar{\eta}) \subset \bar{X}(0;l)$$

$$(A, \sigma)$$

Намагниченность спинового стекла

$$\bar{\eta}_0 = \bar{h}_0 + \sum_{k \in A_0}^{k \neq 0} s_k \bar{h}_k = \sum_{i \in X_0} n_i \bar{h}_i \quad s_k = \prod_{a \in C_k^{(0)}} \sigma_a$$

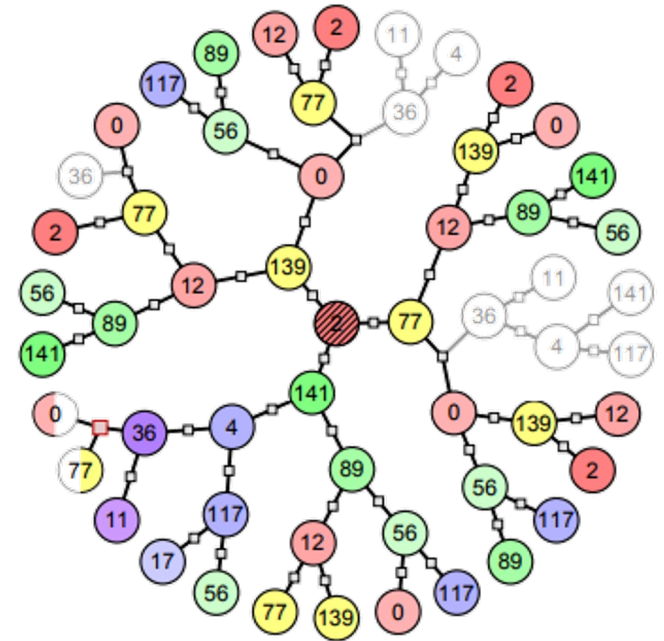
TS(a,b), b ≠ 0

Квази-инстантоны (субоптимальные решения в результате неполноты ранга **решаемой системы, циклов в графе**)

$$\frac{\partial}{\partial \bar{h}_j} \sum_{i \in X_0} \left(\frac{s^2 (\bar{h}_i - 1)^2}{2} - \lambda n_i \bar{h}_i \right) = 0; \sum_{i \in X_0} n_i \bar{h}_i = 0 \quad m_j = s^2 \bar{\eta}_0$$

$$\bar{h}_j = 1 - n_j \left(\sum_{k \in X_0} n_k \right) \left(\sum_{k \in X_0} n_k^2 \right)^{-1} \quad \bar{S}(n) = \frac{1}{2} \left(\sum_{k \in X_0} n_k \right)^2 \left(\sum_{k \in X_0} n_k^2 \right)^{-1}$$

Вместо глобального минимума множество локальных



Дерево квазициклического Низкоплотного кода Таннера [155,64,20] для r=4 слоев (после 4 итераций)

Ландшафт потерь квантовых моделей на калибровочном поле заданном Низкоплотностным кодом.
 “Гладкость” (хорошая обусловленность), возникает при улучшение спектров связности (* устранения треппин-сетов TS(a,b)) и максимизации кодового расстояния TS(a,0)

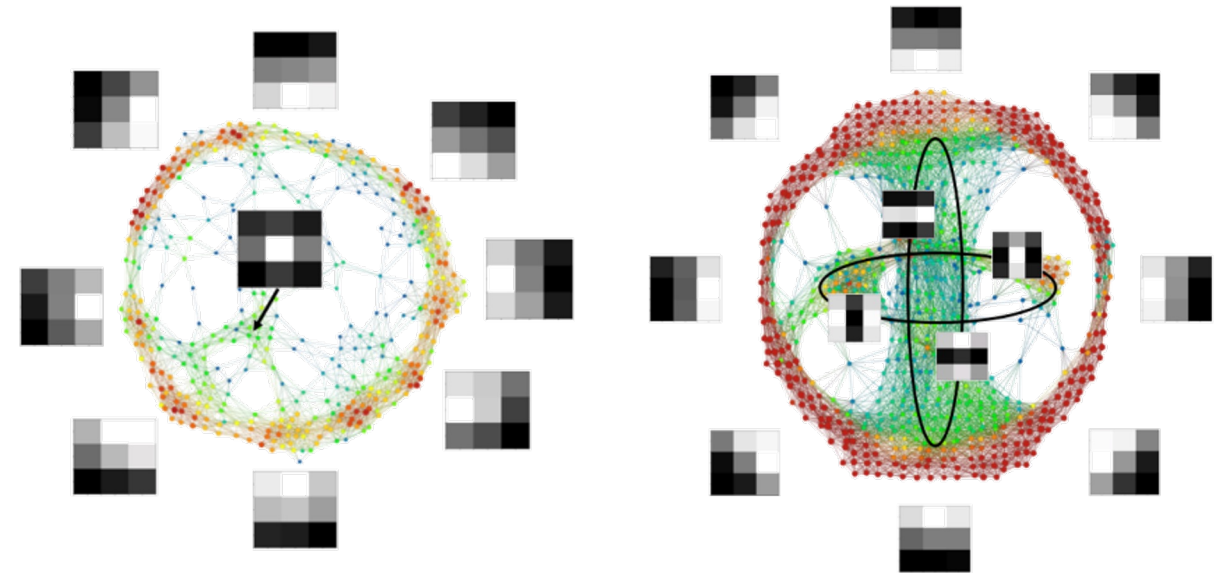


Figure 6: CIFAR-10 layer 2, gray scale Figure 7: First layer, CIFAR-10.

*Ландшафты поверхностей образуются в результате формирования ячеек вокруг кодовых слов TS(a,0) и псевдокодовых слов (TS(a,b)) образованных циклами

** Топология структуры (комплекс) активации тензоров глубокой нейронной сети (CNN, 3x3), обученной на CIFAR-10

*Beom Jun Kim Performance of networks of artificial neurons: The role of clustering. PHYSICAL REVIEW E 69, 045101(R) (2004)

** [1712.09913] Visualizing the Loss Landscape of Neural Nets (arxiv.org)

***Carlsson, G., Gabrielsson, R.B. (2020). Topological Approaches to Deep Learning. In: Baas, N., Carlsson, G., Quick, G., Szymik, M., Thaule, M. (eds) Topological Data Analysis. Abel Symposia, vol 15. Springer

Ошибка реконструкции индивидуального спина лучше при максимизации длины циклов в графе –

Улучшения спектров связности (устранения треппин-сетов $TS(a,b)$) и кодового расстояния $TS(a,0)$

Обученные нейронные сети с высокой точной, высокой обобщающей способностью и низкой вероятностью ложного срабатывания отвечают критериям улучшения спектра связности помехоустойчивых кодов с мягким декодирование:
Методам построения низкоплотностных кодов

*Beom Jun Kim Performance of networks of artificial neurons: The role of clustering. PHYSICAL REVIEW E 69, 045101(R) (2004)
Carlsson, G., Gabrielsson, R.B. (2020). Topological Approaches to Deep Learning. In: Baas, N., Carlsson, G., Quick, G., Szymik, M., Thaulé, M. (eds) Topological Data Analysis. Abel Symposia, vol 15. Springer

Исправление “отравления” данных (adversarial attack)

Нецелевая состязательная атака (adversarial attack, “отравления” данных)

Цель нецелевой состязательной атаки (untargeted adversarial attack, отравления данных) — создать небольшое возмущение во входном изображении, которое заставит нейронную сеть ошибочно отнести его к какому-либо неправильному классу.

Алгоритм нецелевого отравления данных:

1. Вычислите градиент потерь по отношению к входным данным:

$$\mathbf{g} = \nabla_{\mathbf{x}} \ell(\mathbf{x}, y, \theta)$$

2. Вычислить возмущение, взяв знак градиента:

$$\eta = \varepsilon \cdot \text{sign}(\mathbf{g})$$

3. Осуществить отравление данных:

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \eta$$

где $\mathbf{x} \in \mathbb{R}^{n \times m \times 3}$ - входное RGB изображение размера $n \times m$, $y \in \{1, 2, \dots, C\}$ – метка класса, $f(\mathbf{x}, \theta)$ - нейронной сеть с параметрами θ , которая выводит вектор вероятностей классов, $\ell(\mathbf{x}, y, \theta)$ - функция потерь, которая измеряет несоответствие между предсказанными вероятностями классов и истинной меткой класса, ε - величина возмущения, небольшой положительный скаляр, управляющий интенсивностью возмущения,

$$+1, x > 0$$

$\text{sign}(x)$ – функция знака, $\text{sign}(x) = 0, x = 0$.

$$-1, x < 0$$

X
Класс: маргаритка

η

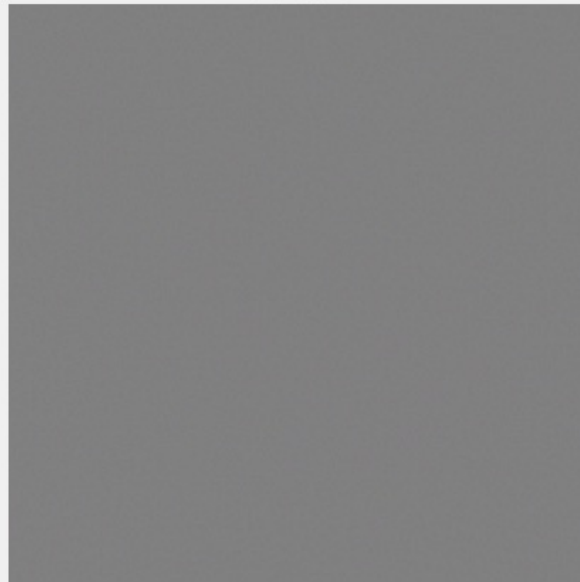
X_{adv}
Класс: конский каштан



Original Image
Class: daisy



Perturbation



Adversarial Image (Epsilon = 1)
Class: buckeye



X
Класс: чайник

η

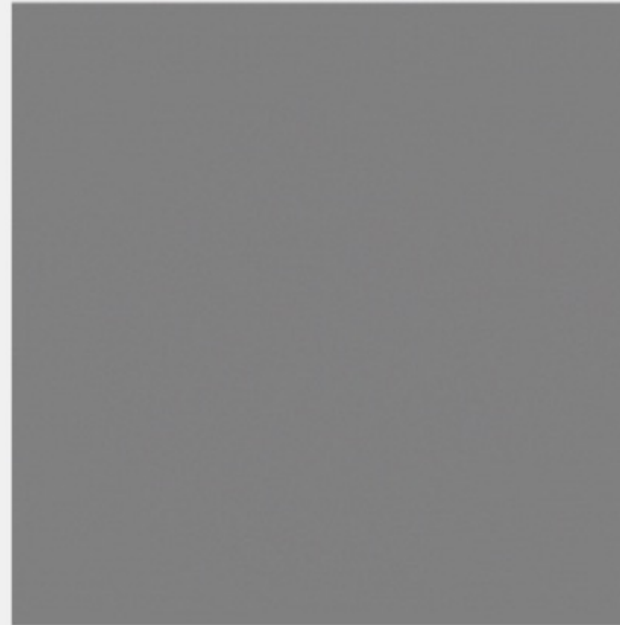
X_{adv}
Класс: кувшин

Original Image
Class: teapot



+

Perturbation



=

Adversarial Image (Epsilon = 1)
Class: pitcher



Исправление состязательной атаки (One shot defence from adversarial attack)

Алгоритм:

1. Преобразование состязательного изображения в оттенки серого:

$$\mathbf{x}_{\text{gray}} = \text{rgb2gray}(\mathbf{X}_{\text{adv}}), \mathbf{X}_{\text{adv}} \in \mathbb{R}^{H \times W \times C}$$

2. Выборка подблоков. Преобразование изображения в непересекающиеся блоки размером $b \times b$:

$$\mathbf{x}_{\text{block}} = \text{im2col}(\mathbf{x}_{\text{gray}}, [b, b], \text{'distinct'})$$

3. Построение разреженного словаря (Generate Sensing Matrix):

$$\mathbf{A} = \text{BCS_SPL_GenerateProjection}(b, \text{ratio}), \mathbf{A} \in \mathbb{R}^{m \times n}$$

4. Вычисление матрицы реконструкции:

$$\mathbf{y} = \mathbf{A}\mathbf{x}_{\text{block}}$$

5. Инициализация параметров разреженной реконструкции:

$$\text{opts} = \{\mu, \beta, \theta, \lambda, \text{maxit}, \text{tol}, \mathbf{x}_{\text{gray}}, b, \text{ratio}, H, W\}$$

6. Разреженная реконструкция изображения: $\mathbf{x}_{\text{rec}} = \text{RCoS}(\mathbf{A}, \mathbf{b}, \text{opts})$

7. Для каждого канала итеративно обновите восстановленное изображение, вычитая замаскированное опорное изображение на основе QC-LDPC кода заданного проверочной матрицей \mathbf{H} , масштабированное на α :

$$\mathbf{I}_{\text{ref}} = \mathbf{X}_{\text{adv}}, \mathbf{I}_{\text{rec}}[:, :, c] = \mathbf{I}_{\text{ref}}[:, :, c] - (1 - \mathbf{H}) \cdot \alpha \cdot \mathbf{I}_{\text{ref}}[:, :, c]$$

8. Многоэтапное восстановление со сдвигом маски \mathbf{H} , $\mathbf{H}_{\text{shifted}}(i, j) = \mathbf{H}((\text{shift} + i) \bmod W, j)$, R

$$\mathbf{I}_{\text{rec}}[:, :, c] = \mathbf{I}_{\text{rec}}[:, :, c] + \mathbf{H}_{\text{shifted}} \cdot \alpha \cdot \mathbf{x}_{\text{rec}}$$

9. Возвращаем $\mathbf{x}_{\text{defence}} = \mathbf{I}_{\text{rec}}$

Шаг 6. Устойчивое сжатое зондирование (*Robust Compressed Sensing Recovery*), RCoS($\mathbf{A}, \mathbf{b}, \text{opts}$)*

$$\mathbf{x}_{\text{rec}} = \text{RCoS}(\mathbf{A}, \mathbf{b}, \text{opts}) = U:$$

$$\min_U \lambda_1 \| \nabla U \|_1 + \frac{\beta}{2} \| \nabla U - W \|_2^2 + \frac{\mu}{2} \| AU - b \|_2^2 - \langle \sigma_x, \nabla_x U - W_x \rangle - \langle \sigma_y, \nabla_y U - W_y \rangle - \langle \delta, AU - b \rangle$$

где, U результат сжатой реконструкции, $\nabla U = (\nabla_x U, \nabla_y U)$ градиент U , $W = (W_x, W_y)$ переменная для (Total variation denoising)-регуляризации**, $\sigma_x, \sigma_y, \delta$ - множители Лагранжа для ограничений, λ_1, β, μ - регуляризационные параметры.

Алгоритм RCoS:

1. Вычислить $A^T b$ и проинициализировать U, μ, β .
2. Итеративная оптимизация, до достижения maxit. Вычислить градиент $g2$ и связанные с ним параметры.

Выполнить градиентный спуск:

$$g2 = D^T (\nabla U - W), g = A^T (AU - b), U \leftarrow U - \tau \cdot d, d = g2 + \frac{\mu}{\beta} g - D^T sAtd + \theta(U - X)$$

3. Регуляризовать:

$$W_x = \max\left(\left|\nabla_x U - \frac{\sigma_x}{\beta}\right| - \frac{1}{\beta}, 0\right) \cdot \text{sign}\left(\nabla_x U - \frac{\sigma_x}{\beta}\right), W_y = \max\left(\left|\nabla_y U - \frac{\sigma_y}{\beta}\right| - \frac{1}{\beta}, 0\right) \cdot \text{sign}\left(\nabla_y U - \frac{\sigma_y}{\beta}\right)$$

4. Обновить множители Лагранжа $\sigma_x, \sigma_y, \delta$:

$$\sigma_x \leftarrow \sigma_x - \beta(\nabla_x U - W_x), \sigma_y \leftarrow \sigma_y - \beta(\nabla_y U - W_y), \delta \leftarrow \delta - \mu(AU - b)$$

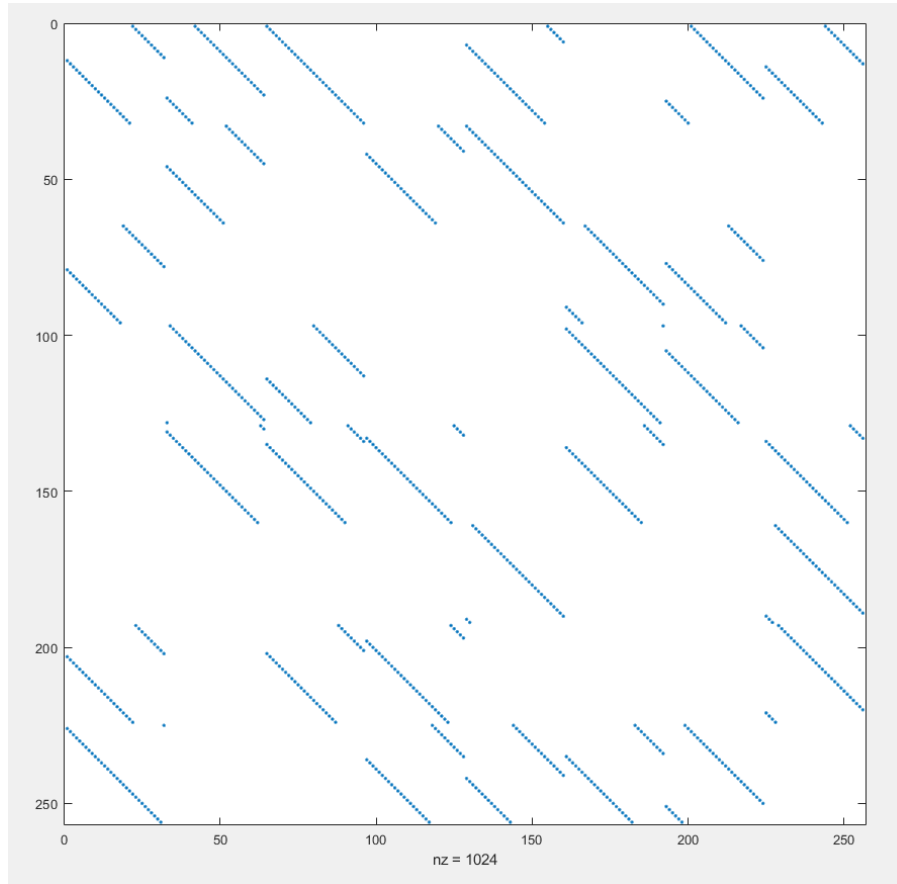
5. Проверить сходимость: $\text{RelChg} = \frac{\|U - U_{\text{prev}}\|_F}{\|U_{\text{prev}}\|_F}$, $\text{RelChg} < \text{tol}$ завершить алгоритм, вернуть U .

*J. Zhang, et al., Image compressive sensing recovery using adaptively learned sparsifying basis via L0 minimization, Signal Processing (2013)

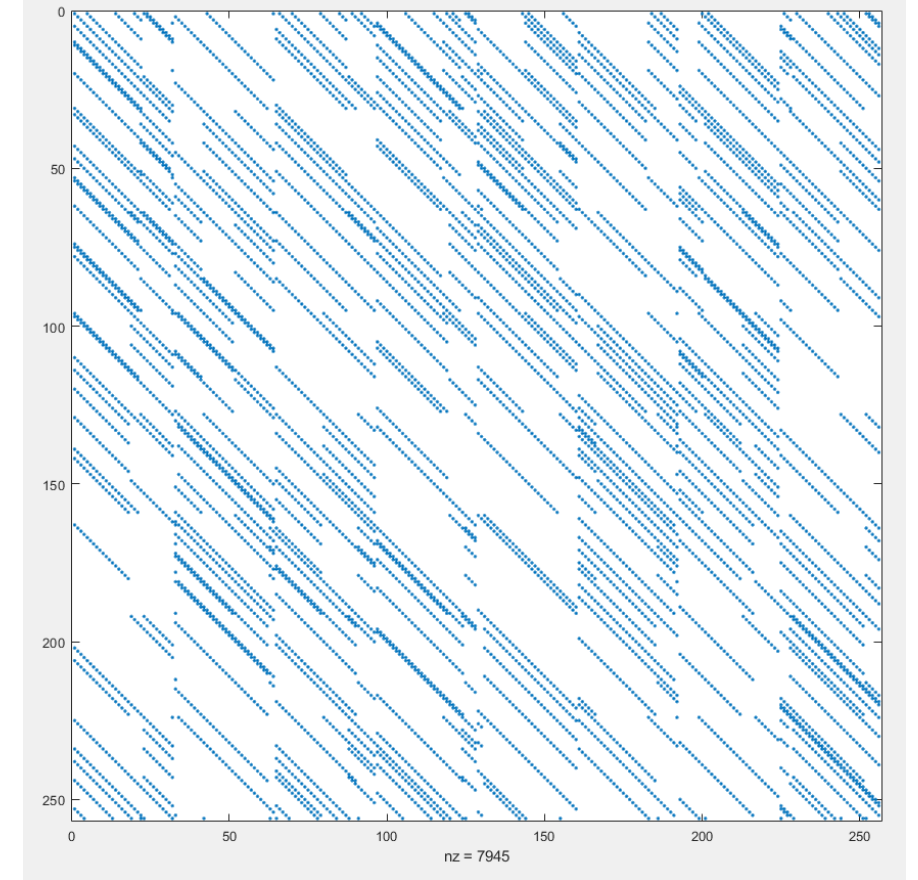
Rudin, L. I.; Osher, S.; Fatemi, E. (1992). "Nonlinear total variation based noise removal algorithms". *Physica D*. **60 (1–4): 259–268.

Шаг 7,8. Применение построенной маски QC-LDPC кода \mathbf{H} и его циклического сдвига

$\mathbf{H}^{256 \times 256} =$



$\mathbf{H}_{\text{shifted}} =$



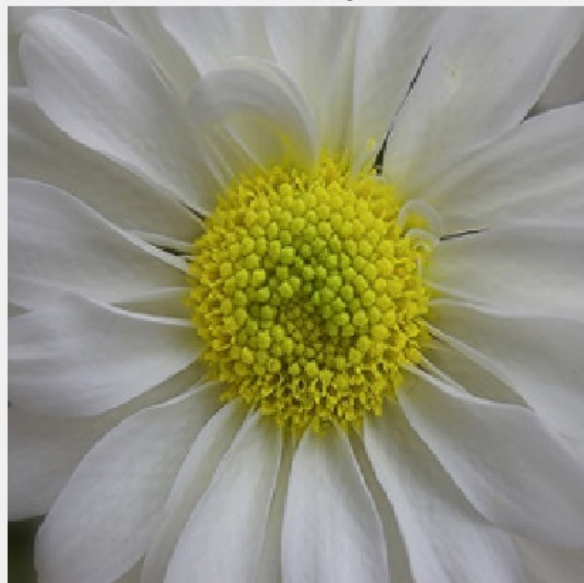
Метод построения \mathbf{H} увеличивает кодовое расстояние и спектр связности*

*:V. S. Usatyuk and S. I. Egorov, "Topology-Aware Sparse Factorization in Energy-Based Models: Tori QC-LDPC Codes and Circular Hyperboloid MET QC-LDPC Codes," *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, Moscow, Russian Federation, 2024, , pp. 1-6
Usatyuk V. S., Egorov S. I., "Heuristic Number Geometry Method for Finding Low-Weight Codeword in Linear Block Codes," *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, Moscow, Russian Federation, 2024, pp. 1-6

X

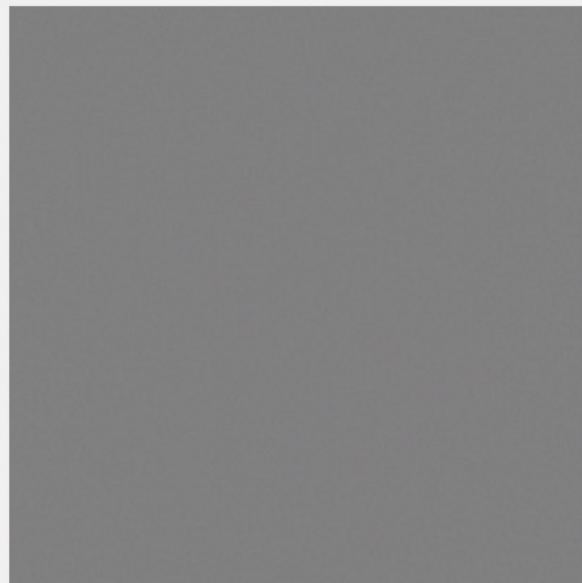
Класс: маргаритка

Original Image
Class: daisy



η

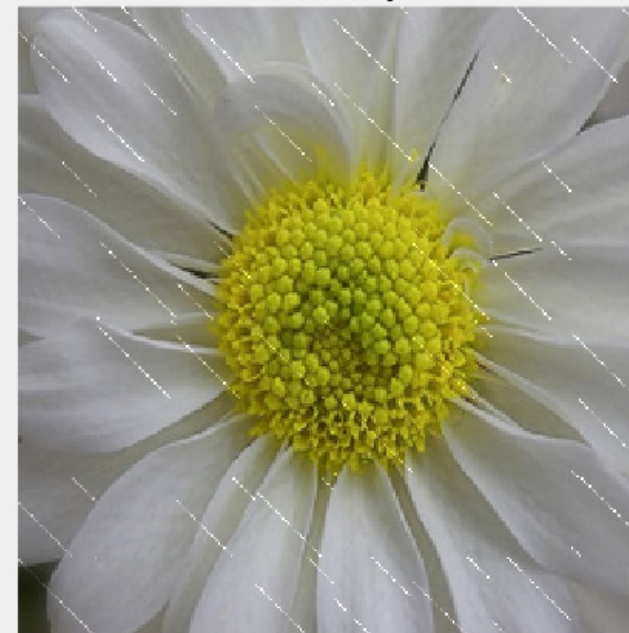
Perturbation



X_{adv}

Класс: маргаритка

Adversarial Image (Epsilon = 1)
Class: daisy



X
Класс: чайник

η

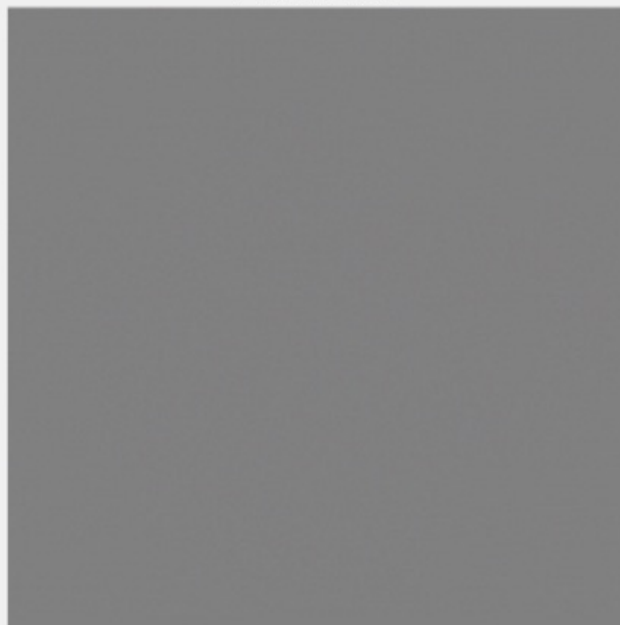
X_{adv}
Класс: чайник

Original Image
Class: teapot



+

Perturbation



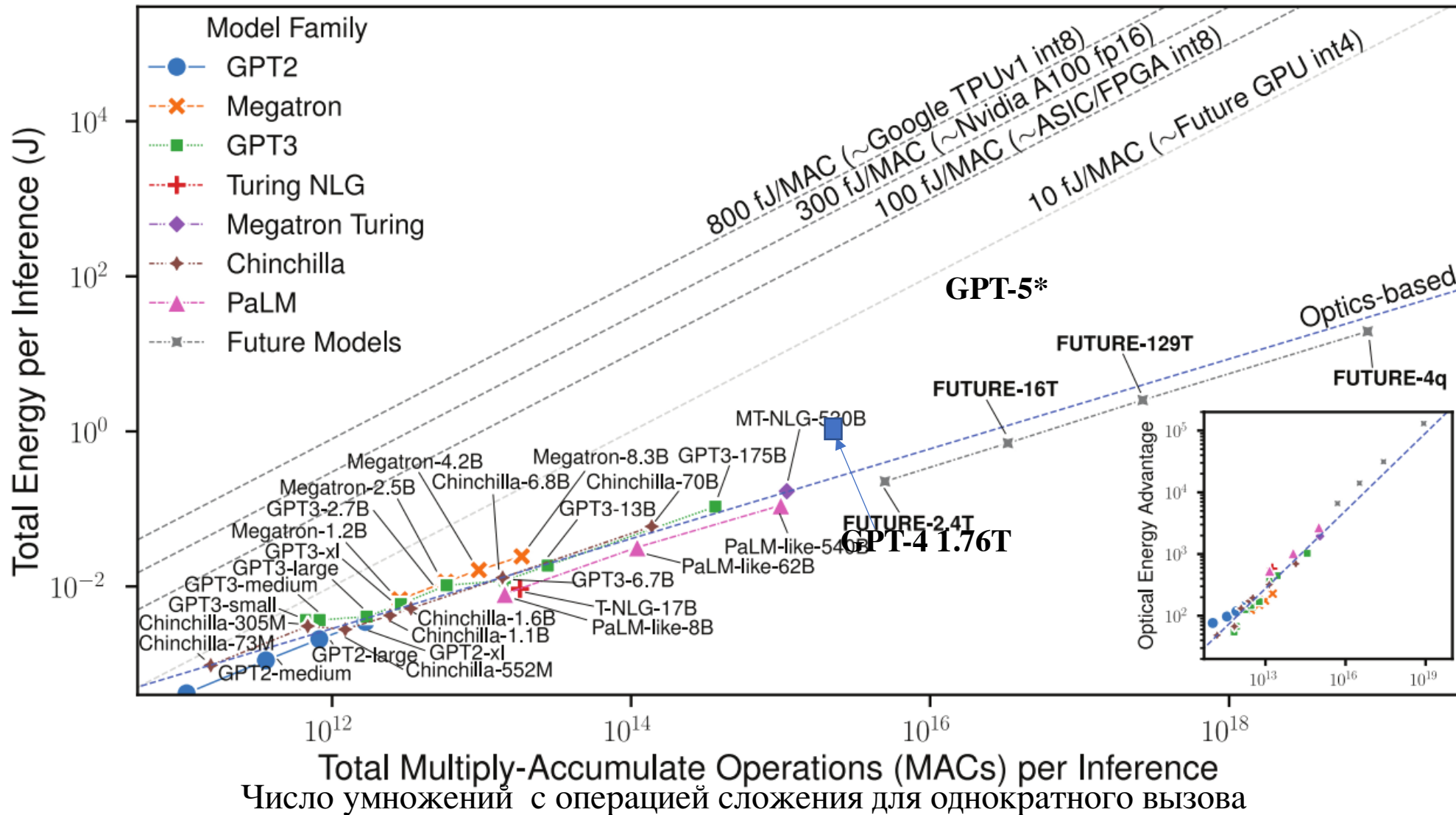
=

Adversarial Image (Epsilon = 1)
Class: teapot



Сложность Глубоких сетей типа трансформер (в том числе визуальные трансформеры)

Энергия в Джоулях для применения Глубокой нейронной сети (Трансформер)



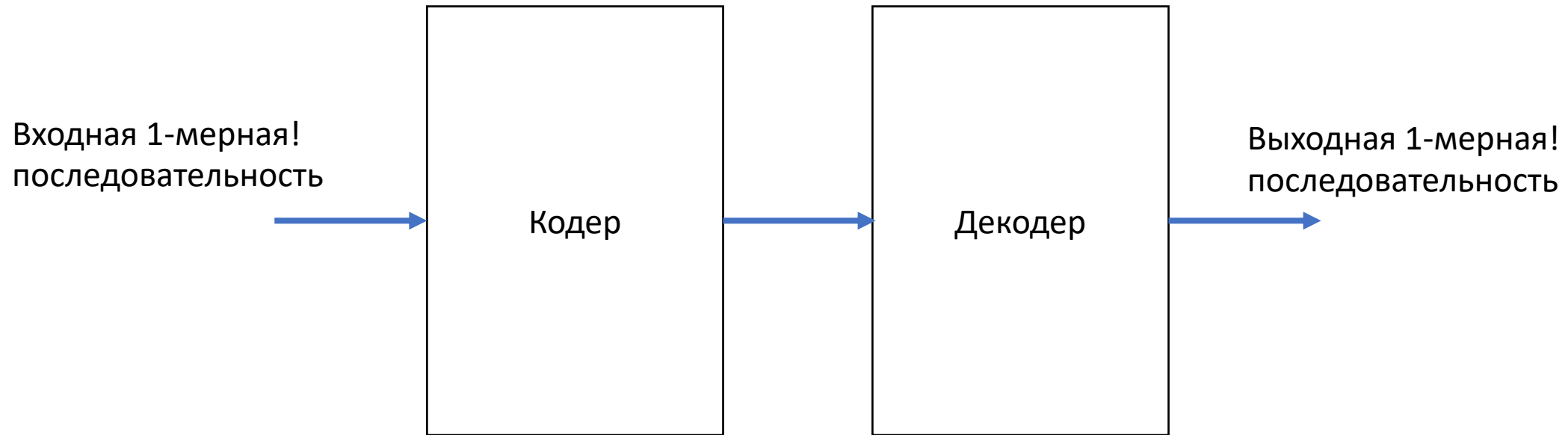
GPT-5*
Потребуется
30 пета-МАС
(*,+)

Текущий уровень типовой ВУ 20-40 ТФЛОПС на массовых устройствах, требуется ускорение > на 3 порядка.

[Maxwell Anderson](#), [Shi-Yuan Ma](#), [Tianyu Wang](#), [Logan Wright](#), [Peter McMahon](#) Optical Transformers Published in Transactions on Machine Learning Research (03/2024)

have around 10 times the parameter count of GPT-4.

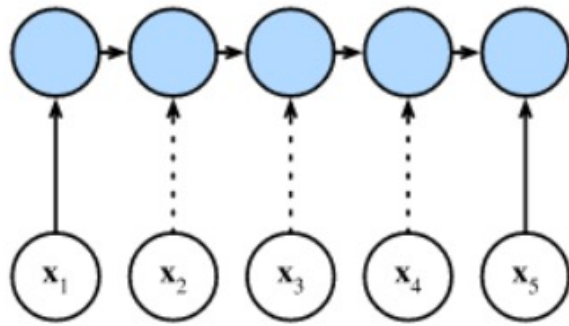
Трансформер



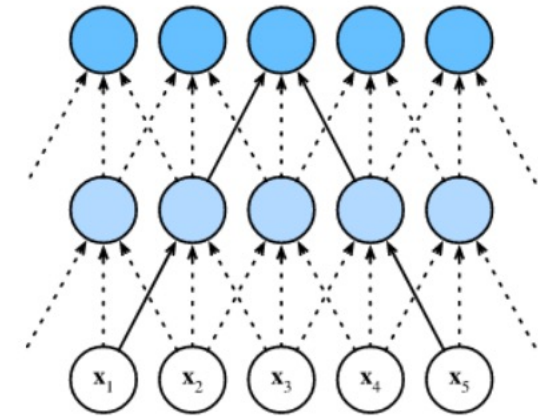
Все входные данные как язык!!!Изображение это reshape многомерных данных, где семантика многомерная есть 1 мерная нелинейная корреляция

Трансформер это общая идея для обработки длинных последовательностей данных

RNN

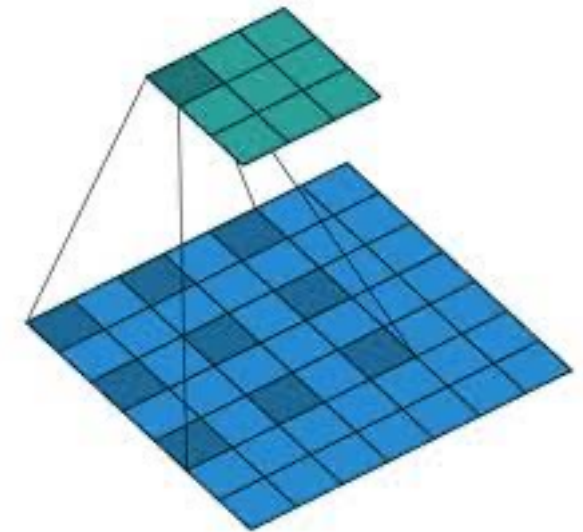
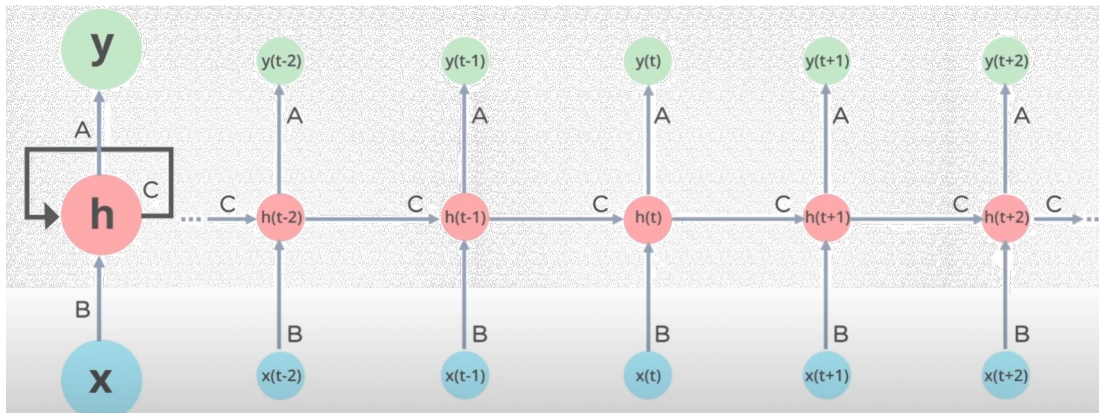


CNN



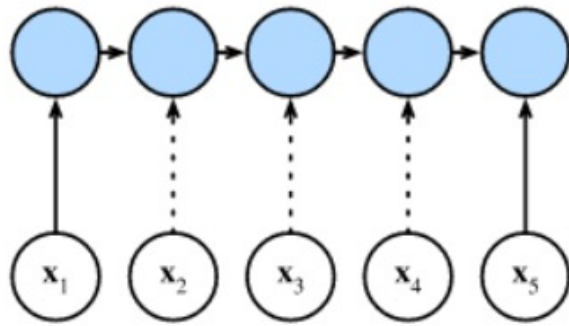
Традиционно для 1D последовательностей применялись Рекуррентные нейронный сети

Традиционно для N-D последовательностей трактуемых как изображения применялись Сверточные нейронный сети

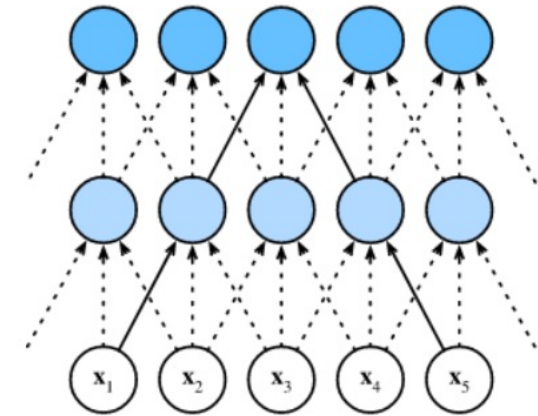


Трансформер это общая идея для обработки длинных последовательностей данных

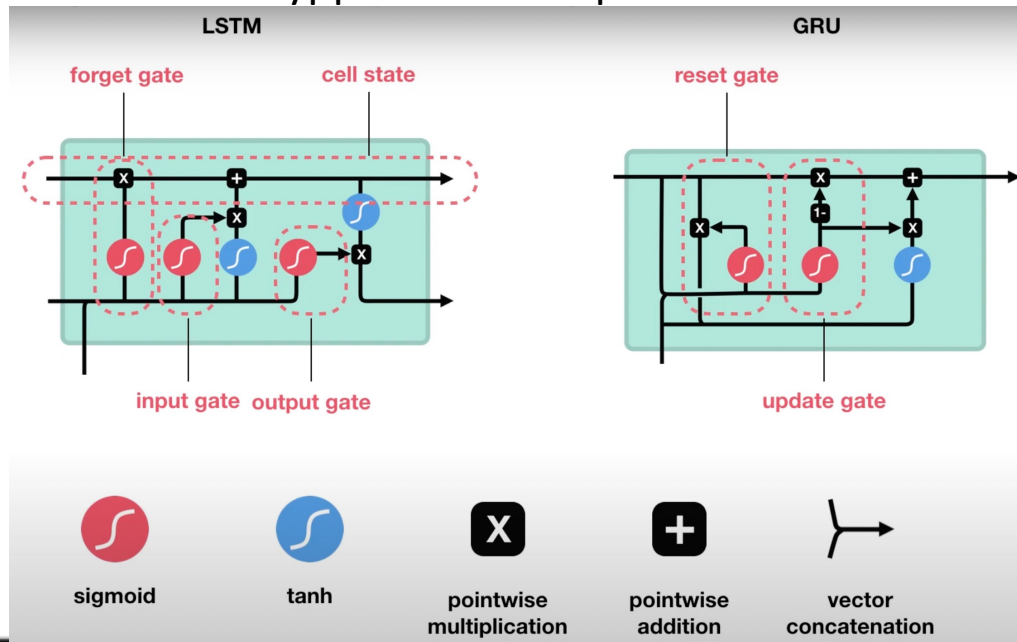
RNN



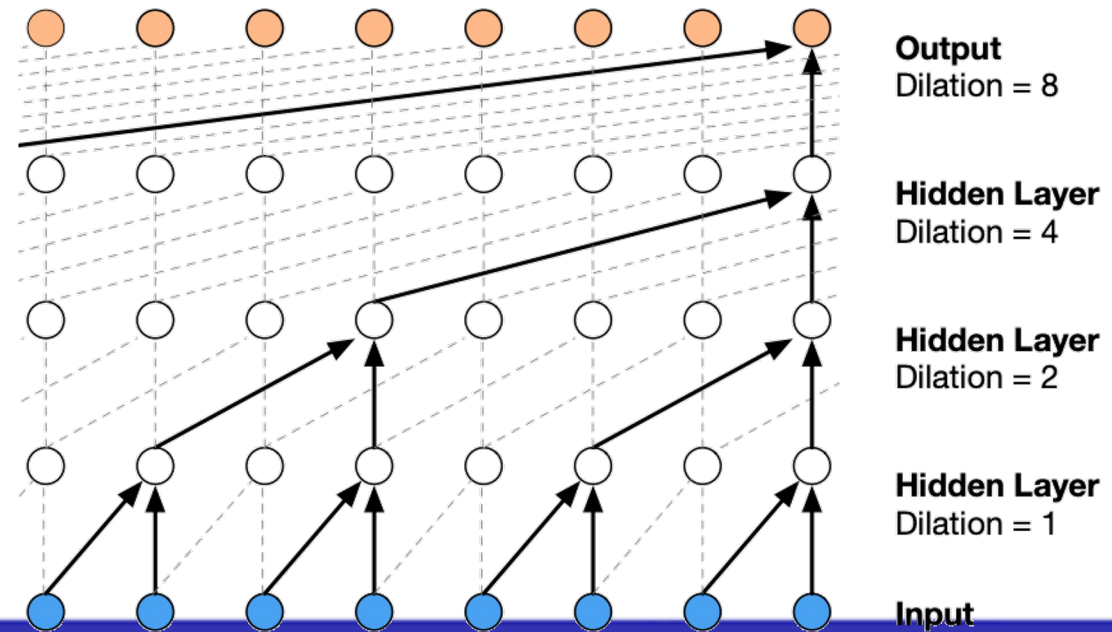
CNN



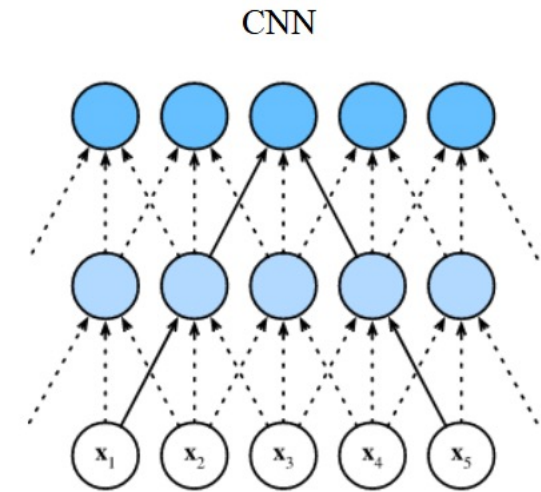
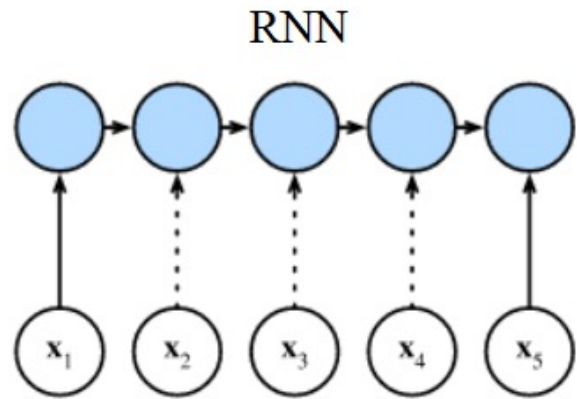
Традиционно для 1D последовательностей применялись Рекуррентные нейронные сети



Традиционно для N-D последовательностей трактуемых как изображения применялись Сверточные нейронные сети



Трансформер это общая идея для обработки длинных последовательностей данных



Однако, обе архитектуры проигрывают трансформерам по сложности и качеству выявления длинных корреляций в данных, в том числе изображениях

Трансформер



Word2token

2339	19081	2024	2488	2084	100
Волга	впадает	в	каспийское	море	?

Трансформер

Кодер

2339 19081 2024 2488 2084 100
Волга впадает в каспийское море ?

Вложение

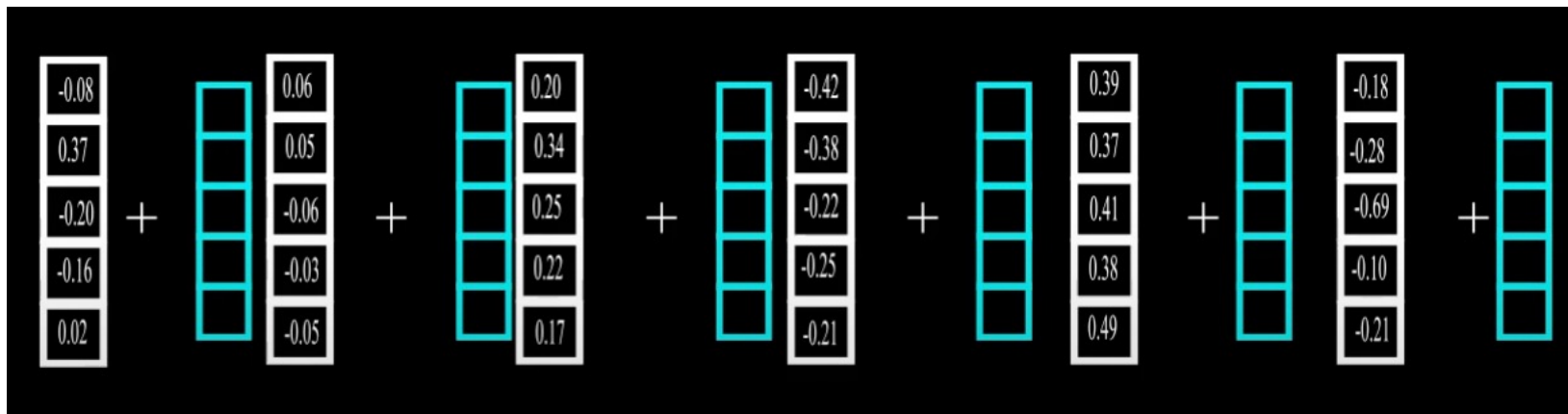
-0.08	0.06	0.20	-0.42	0.39	-0.18
0.37	0.05	0.34	-0.38	0.37	-0.28
-0.20	-0.06	0.25	-0.22	0.41	-0.69
-0.16	-0.03	0.22	-0.25	0.38	-0.10
0.02	-0.05	0.17	-0.21	0.49	-0.21

Трансформер

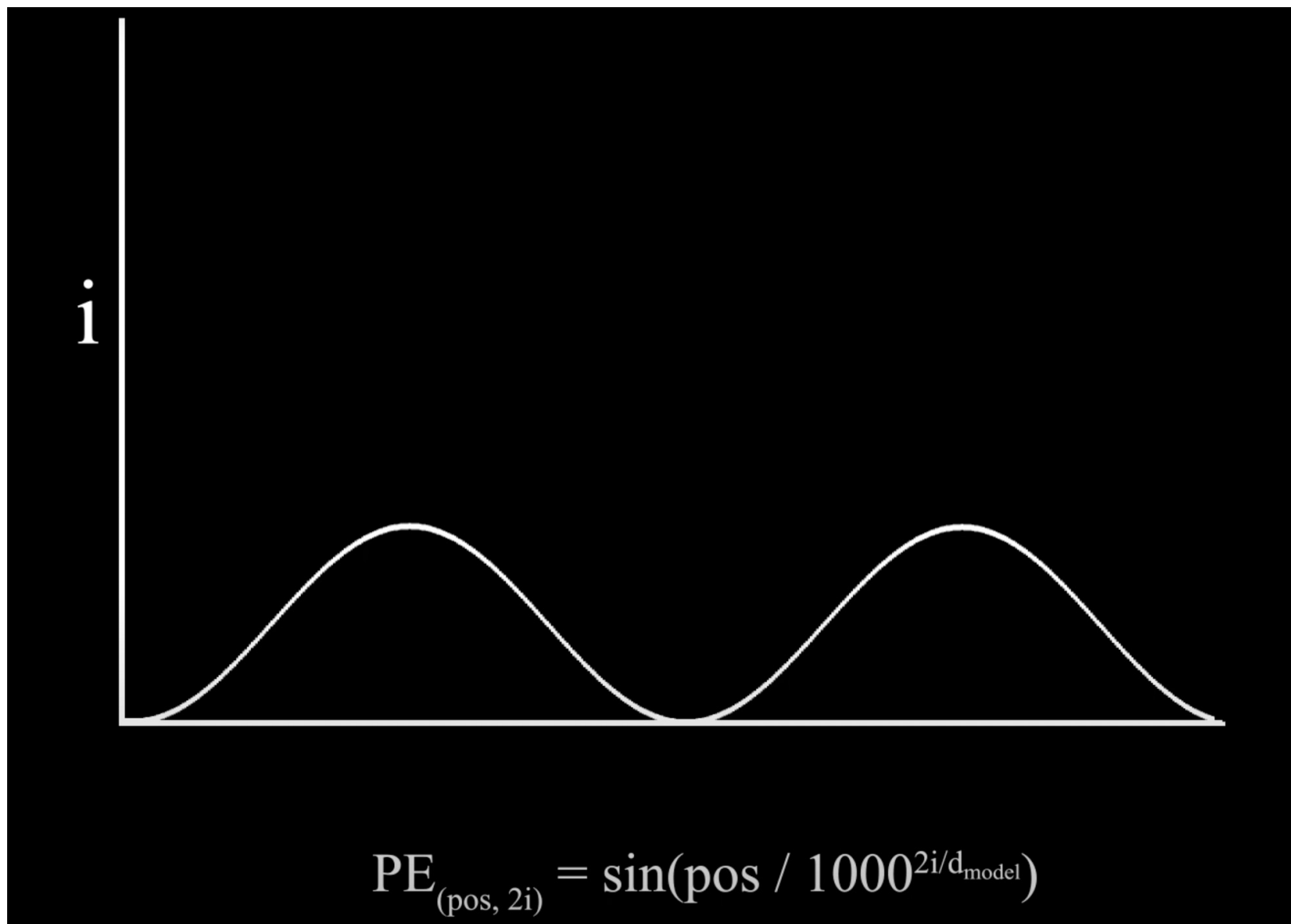
Кодер

2339 19081 2024 2488 2084 100
Волга впадает в каспийское море ?
2488 2084 19081 2024 2339
Каспийское море впадает в Волга?

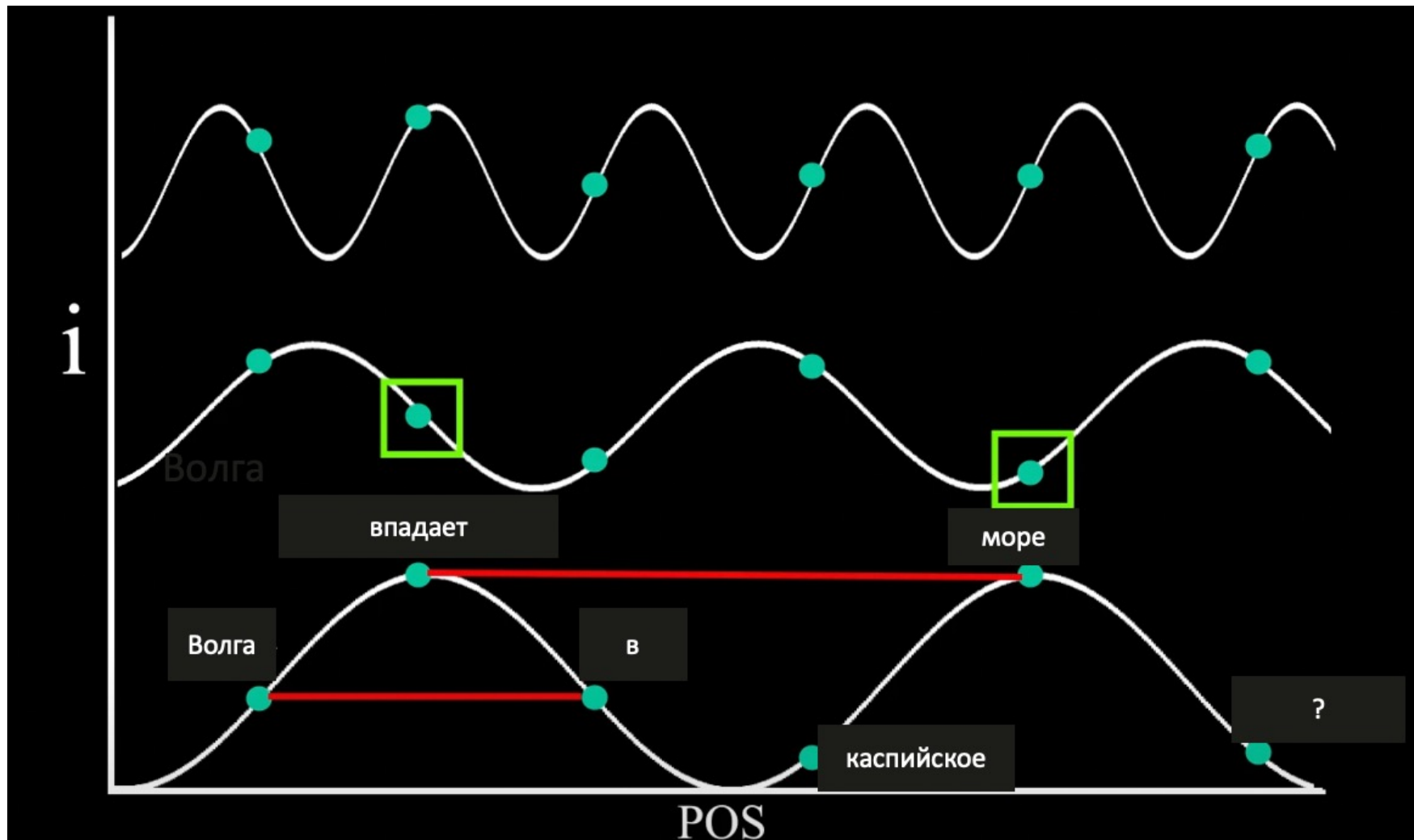
Вложение+ позиционное кодирование
(position vector)



позиционное кодирование (position vector)

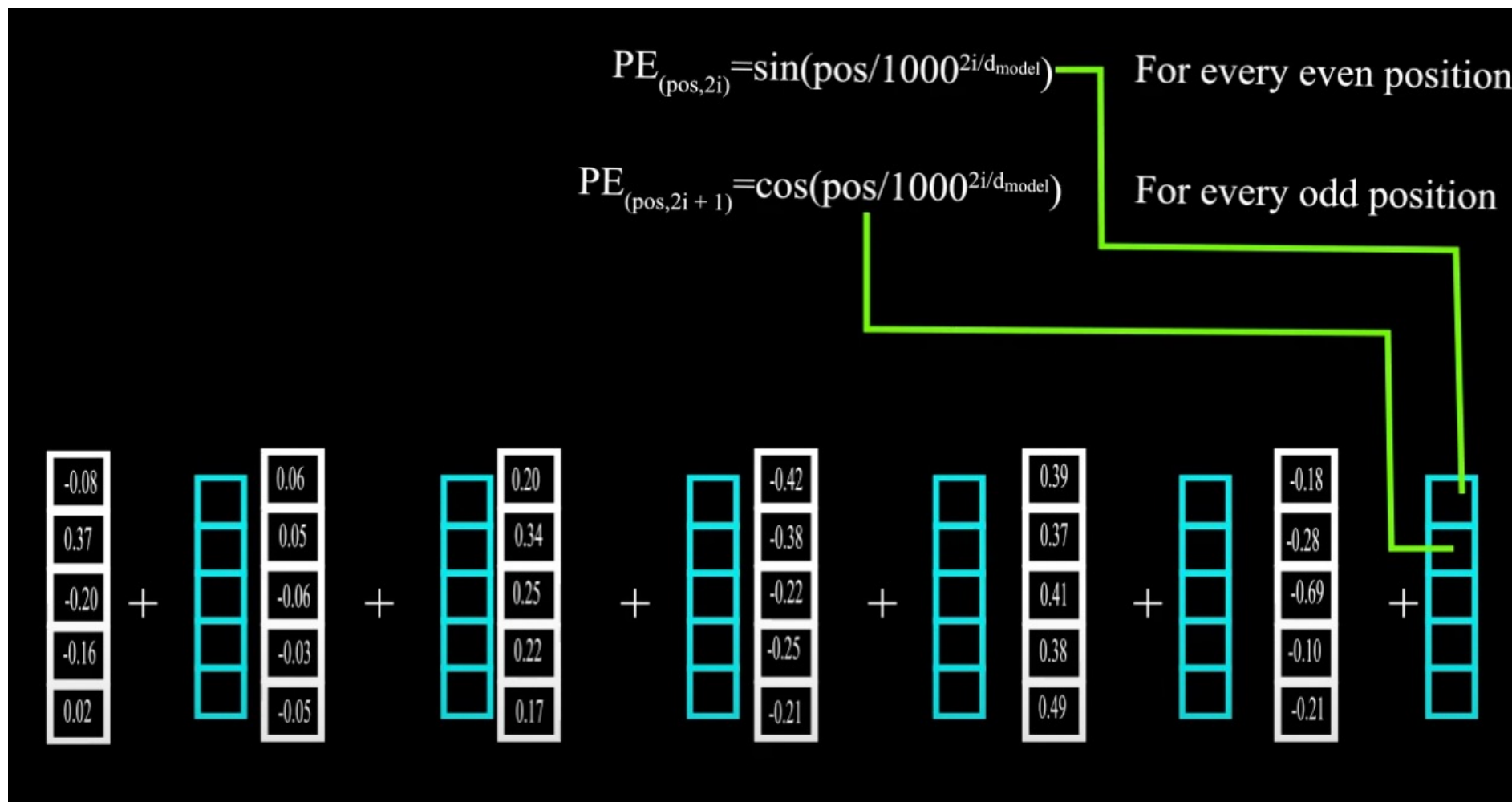


ПОЗИЦИОННОЕ КОДИРОВАНИЕ (position vector)



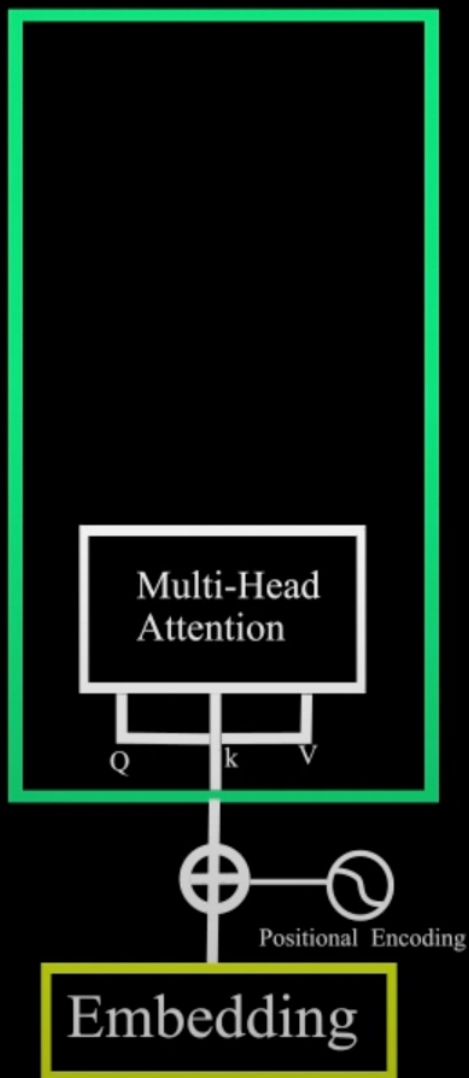
$$PE_{(pos, 2i)} = \sin(pos / 1000^{2i/d_{model}})$$

позиционное кодирование (position vector) с решением проблемы граничных условий



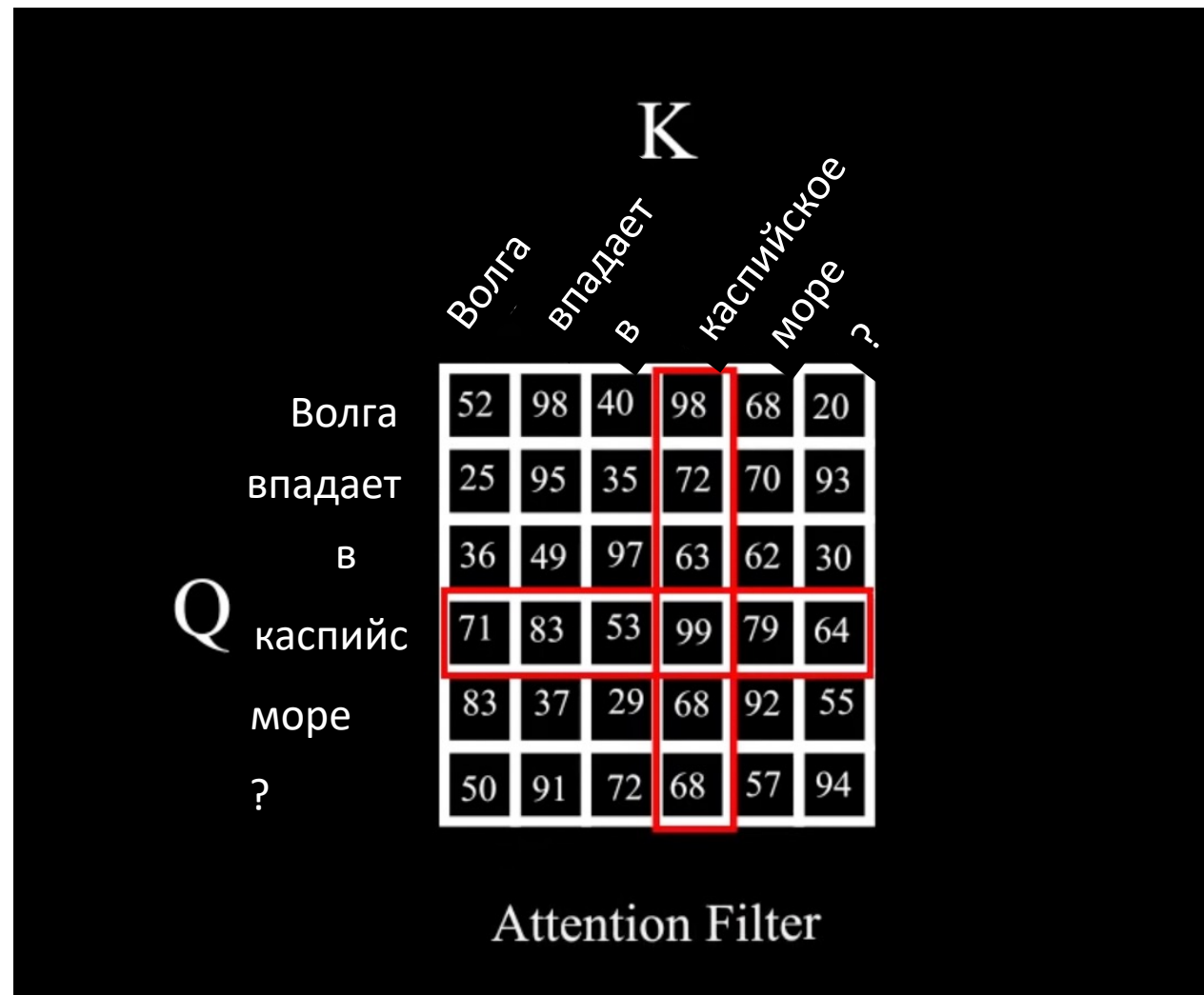
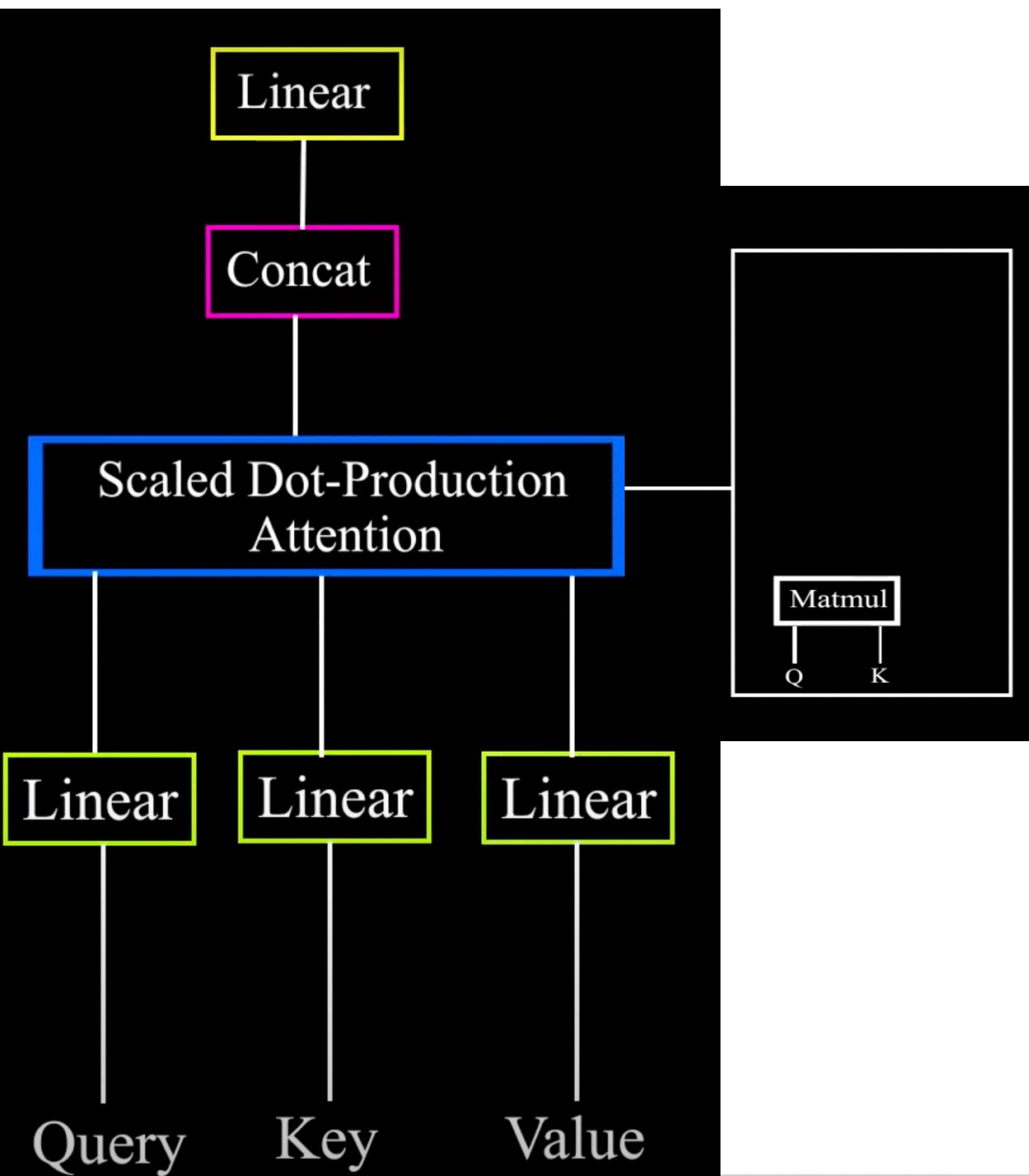
Блок внимания основан на способе хранения информации Query, Key, Value

Запрос: входное значение
Ключ: хеш(свертка)
Значение:



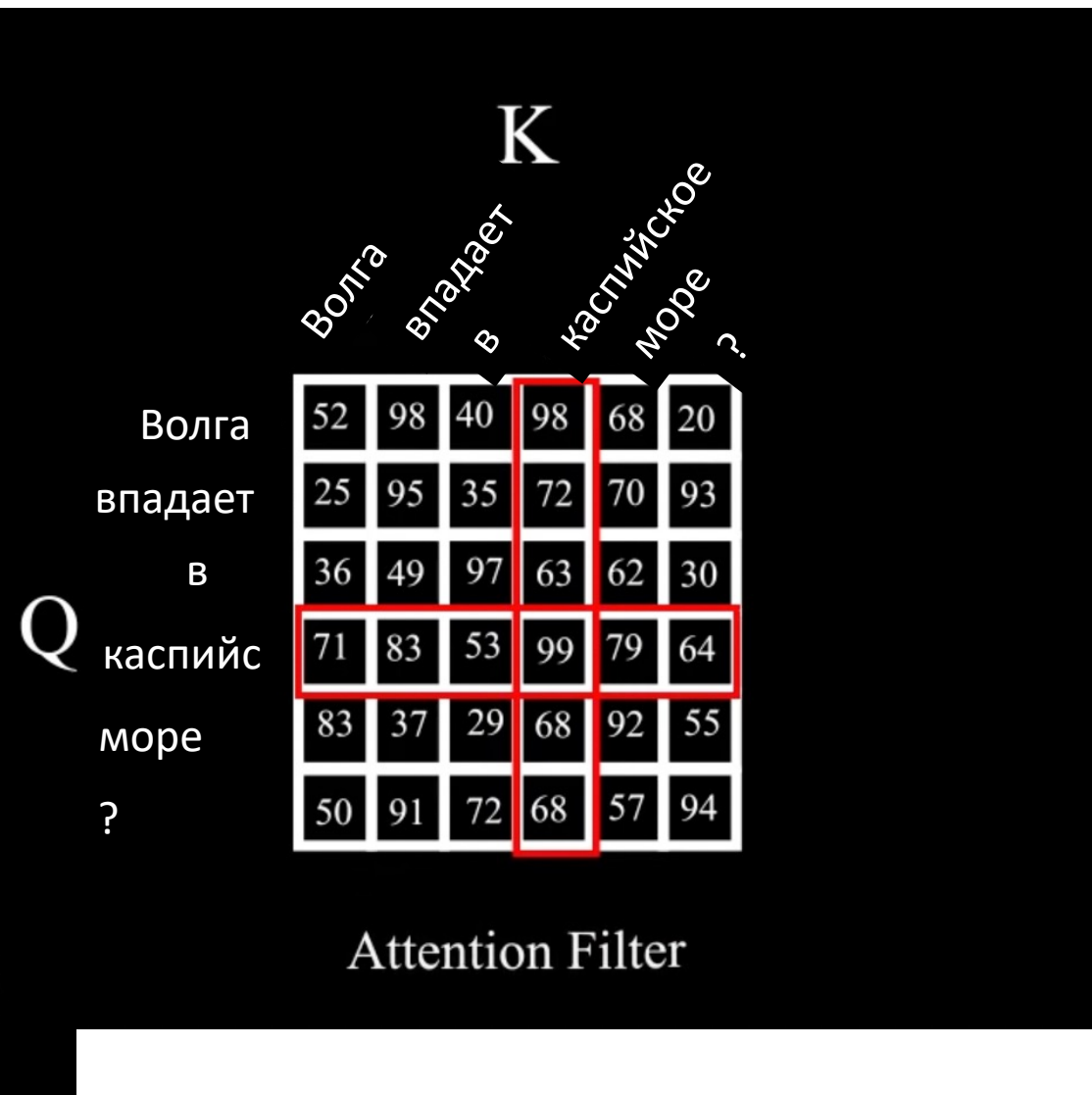
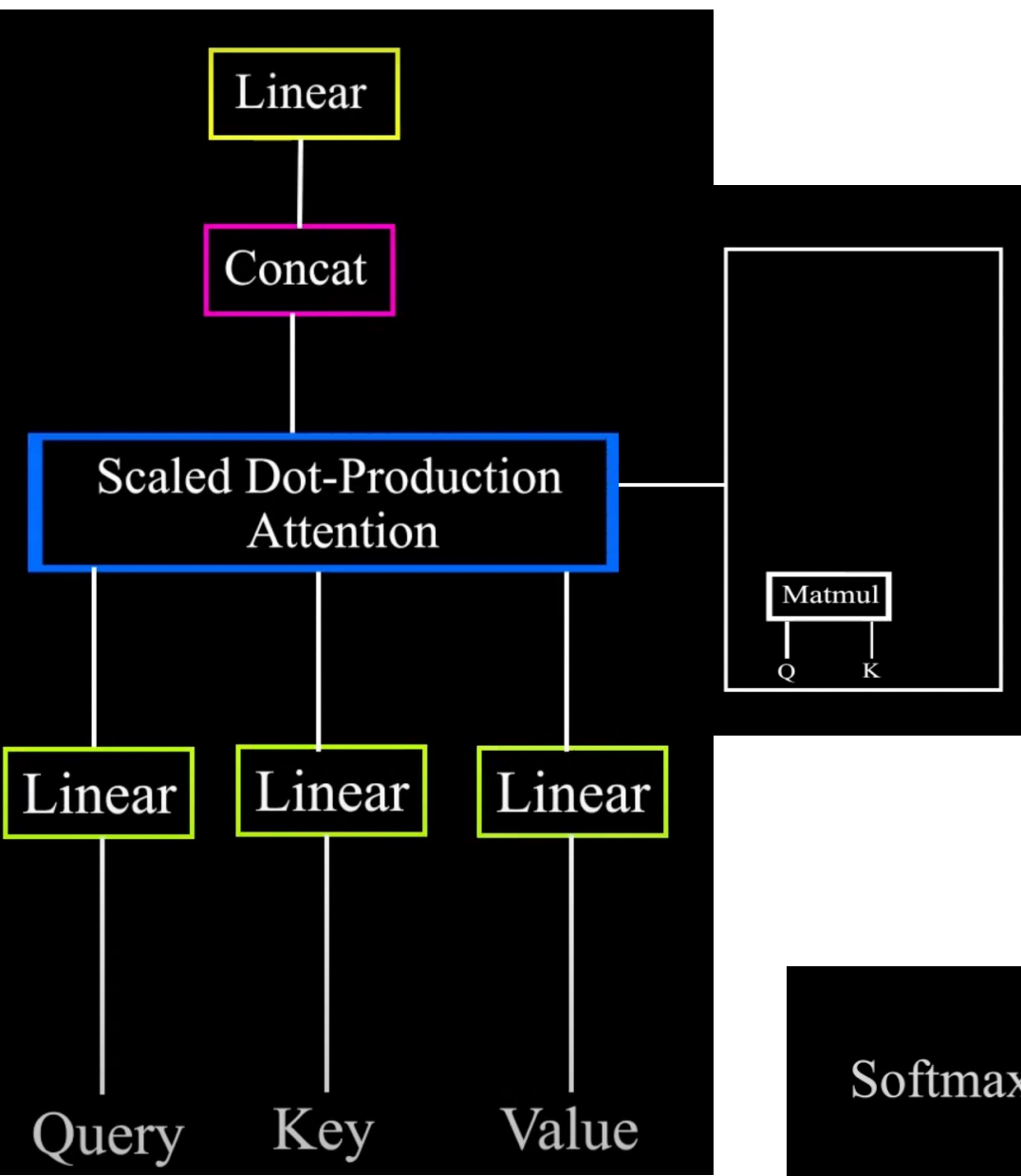
Фильтр внимания внутри головы внимания (Attention head)

Волга впадает в каспийское море ?



Фильтр внимания внутри головы внимания (Attention head)

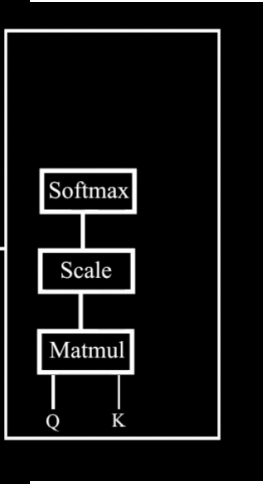
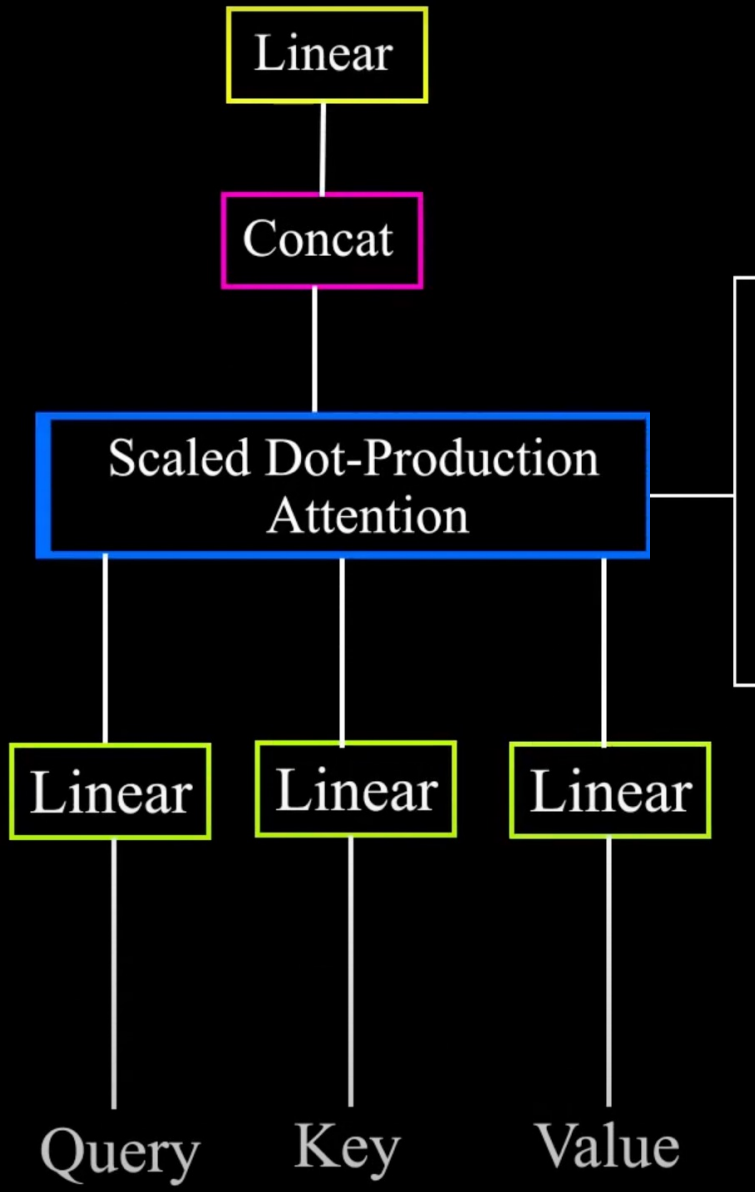
Волга впадает в каспийское море ?



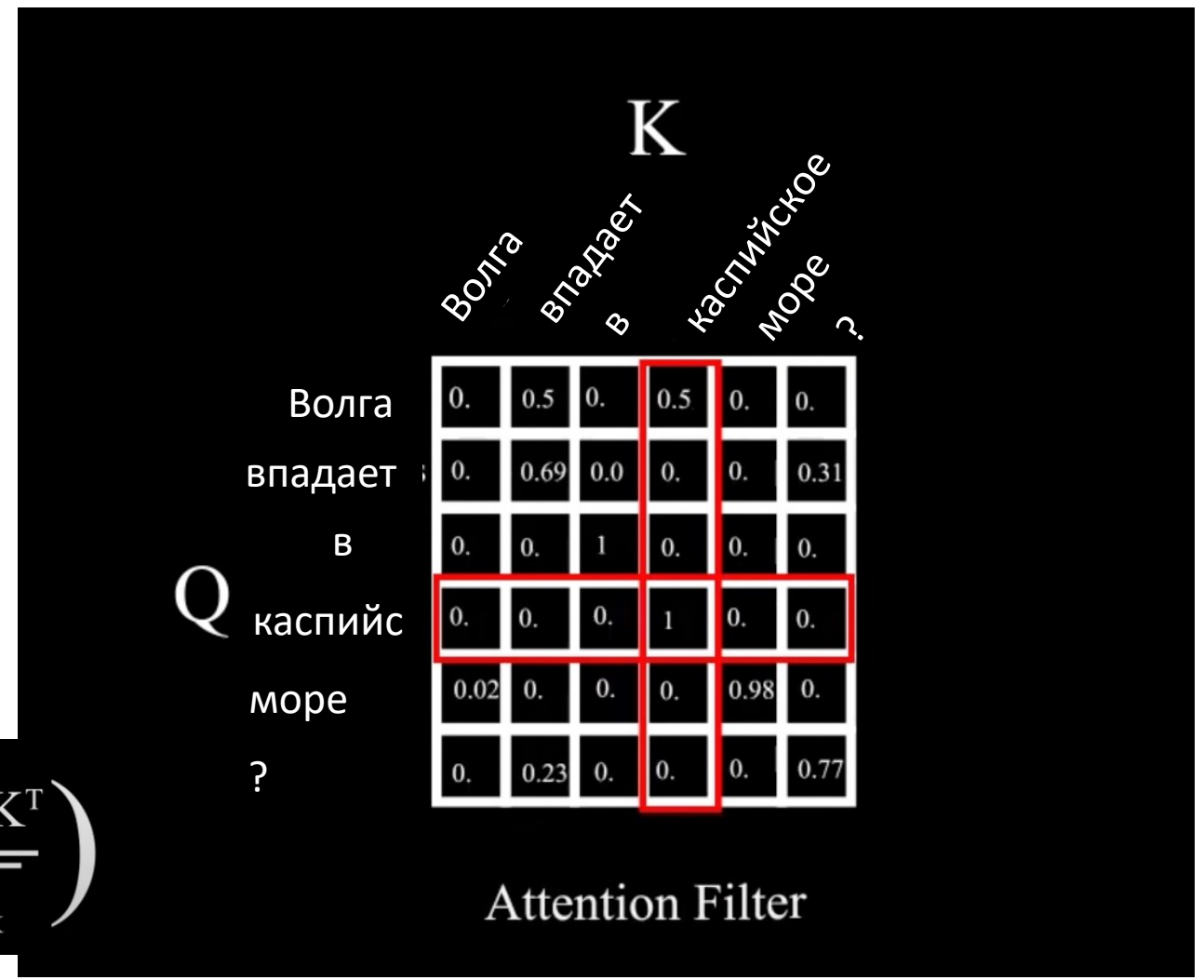
$$\text{Softmax} \left(\frac{Q K^T}{\sqrt{d_k}} \right)$$

Функция активации

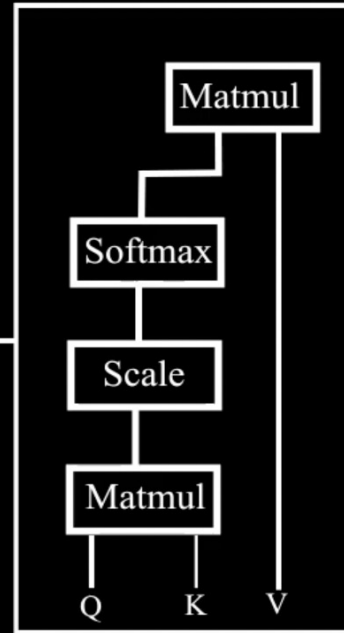
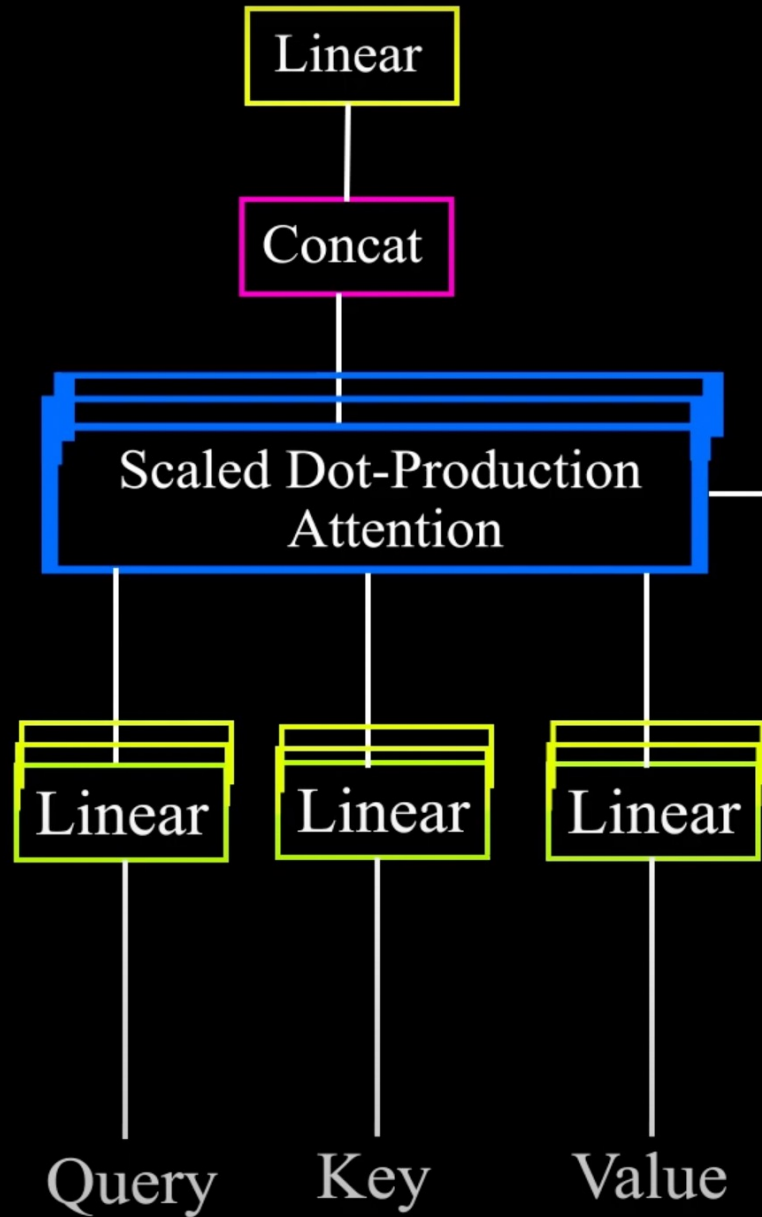
Волга впадает в каспийское море ?



$$\text{Softmax} \left(\frac{Q K^T}{\sqrt{d_k}} \right)$$



Multi-Head Attention



$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{Q K^T}{\sqrt{d_k}} \right) V$$

Multi-Head = Concat(head₁, head₂, ..., head_n)

where head₁ = Attention(QW_i^q, KW_i^k, VW_i^v)

Query L1

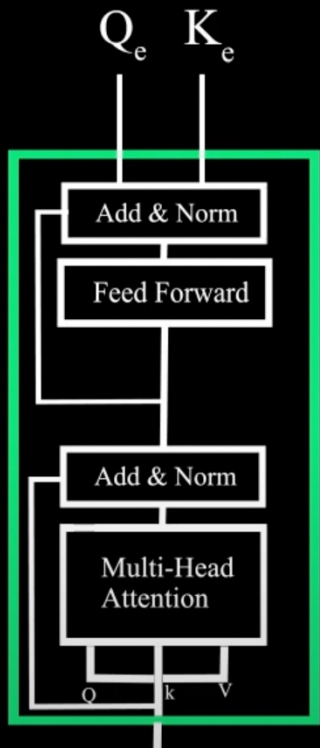
Key L2

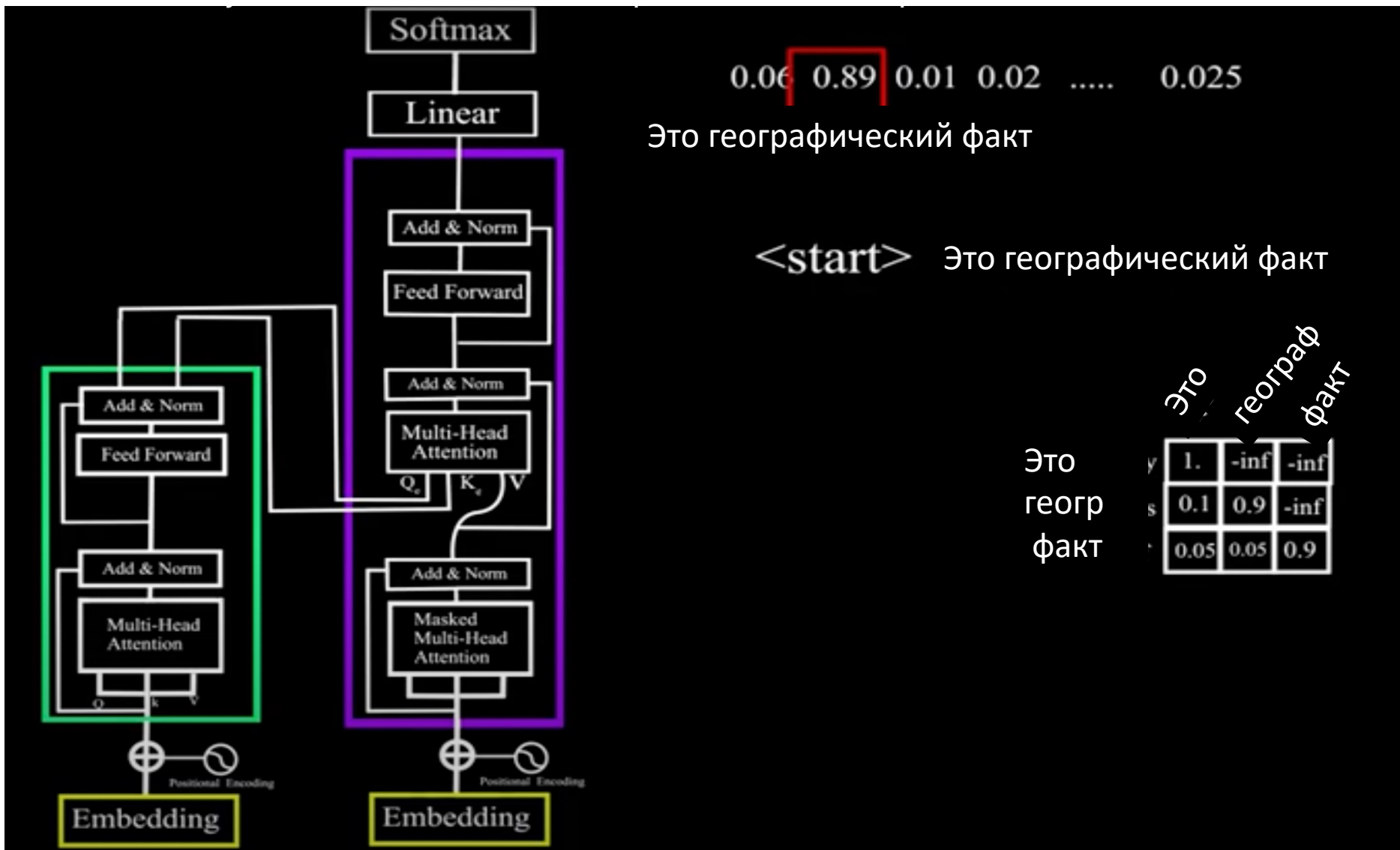
Value L3

Волга впадает в каспийское море ?

Волга впадает в каспийское море ?

Волга впадает в каспийское море ?





0.06 0.89 0.01 0.02 0.025

Это геогра. Факт

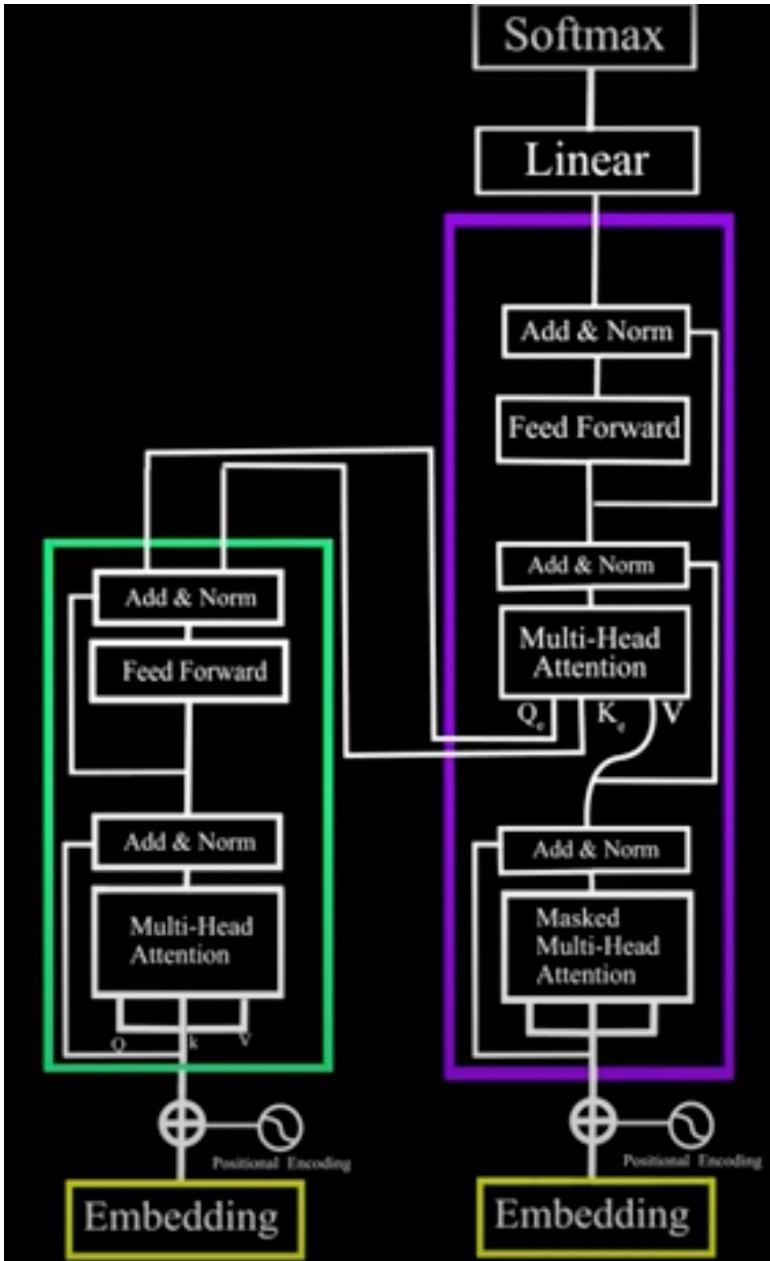
<start> Это географический факт

<end>

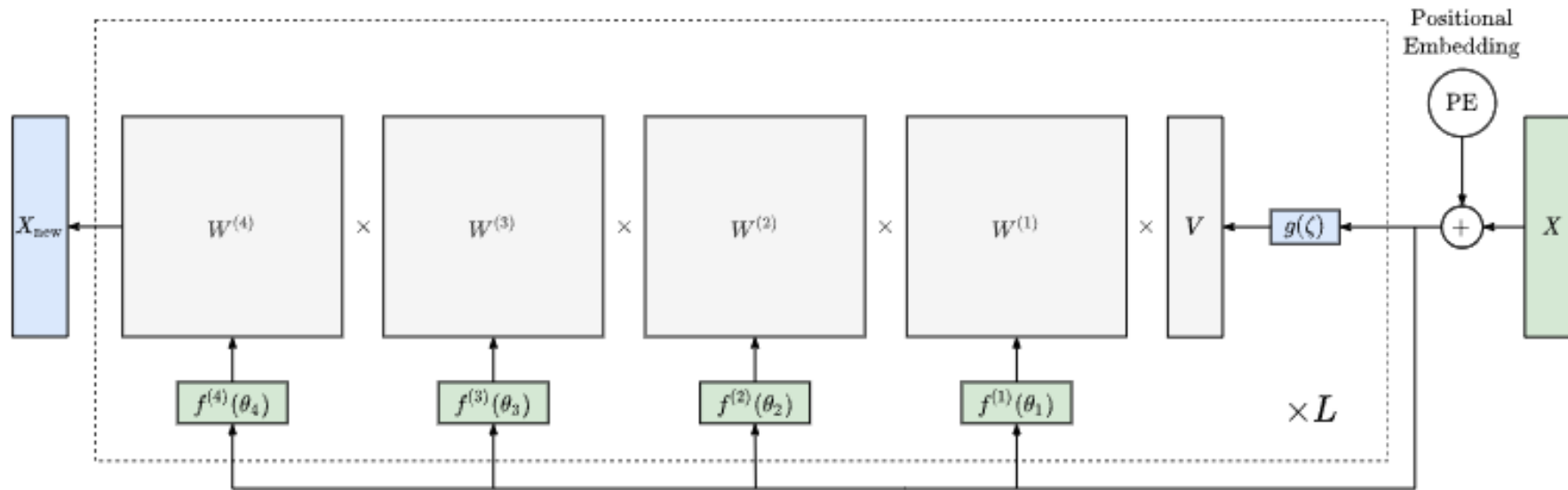
T = 4

Это
геогр
факт

	Это	географ	факт
1.	0.	0.	0.
0.1	0.9	0.	0.
0.05	0.05	0.9	0.
0.01	0.09	0.05	0.



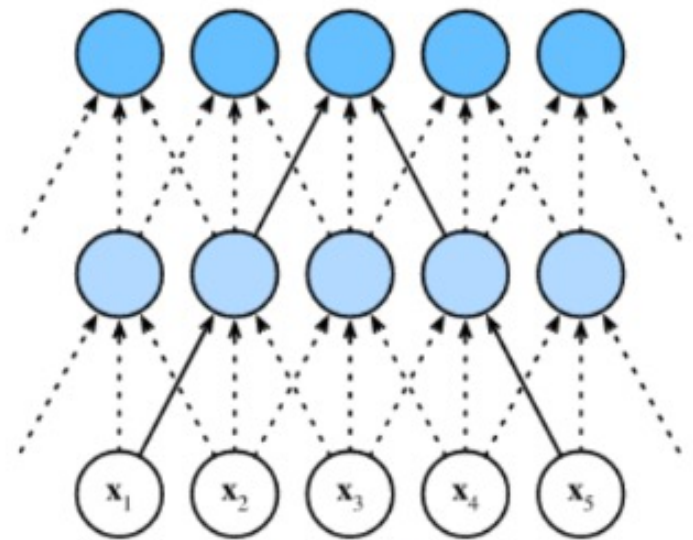
Факторизуем внимание при помощи LDPC кода



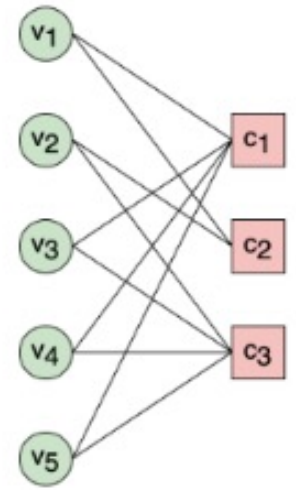
$$\min_{W^{(1)}, \dots, W^{(M)}} \left\| X - \prod_{m=1}^M W^{(m)} \right\|_F^2$$

Архитектуры нейронных сетей и Низкоплотный код для обработки длинных последовательностей данных

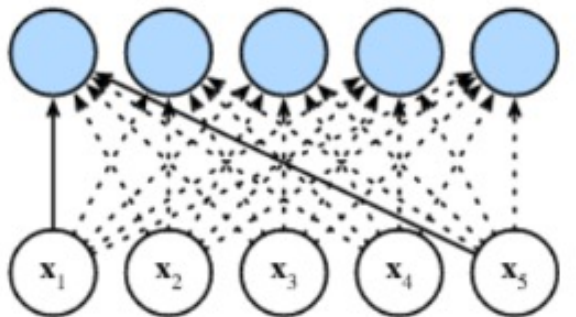
CNN



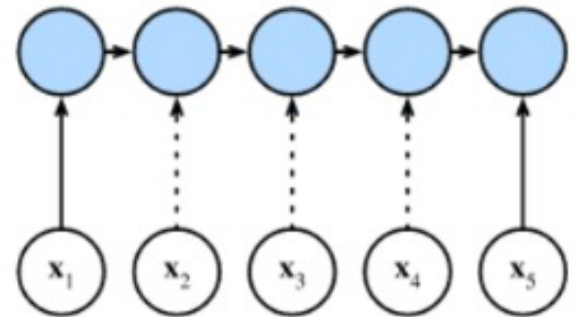
LDPC



Self-attention

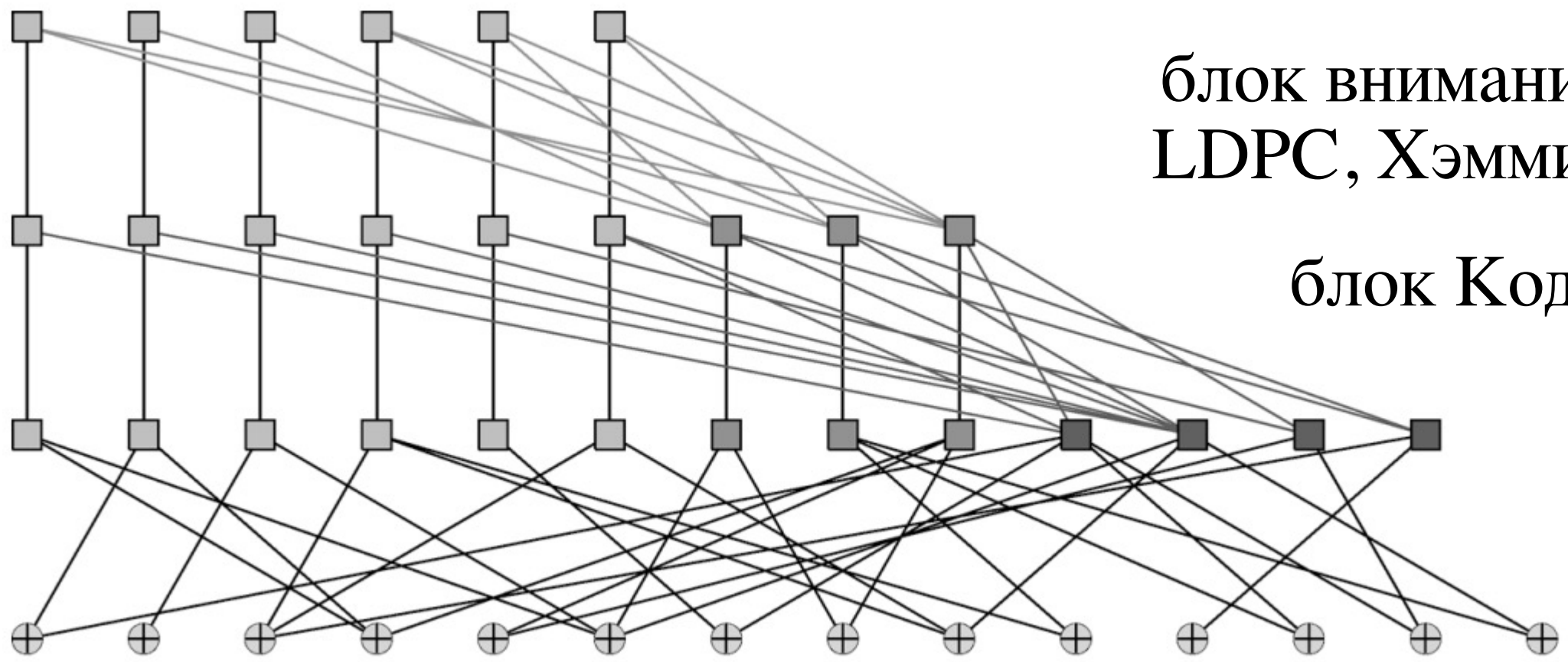


RNN



Факторизовать систему связей произвольной нейронной сети

y_1 ... y_{t-1} y_t Выход



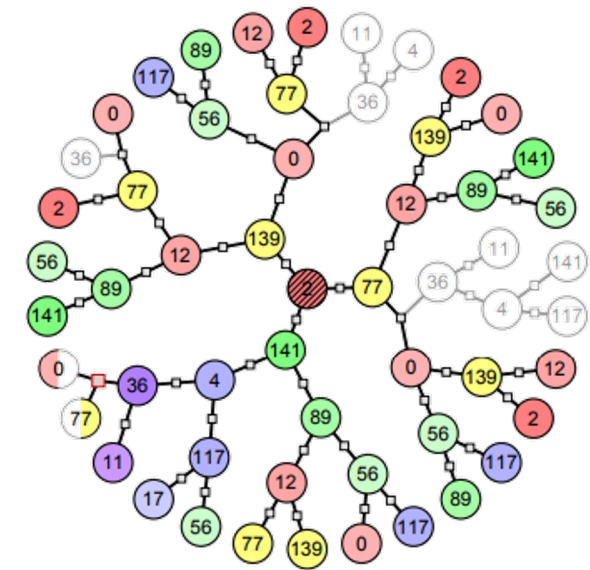
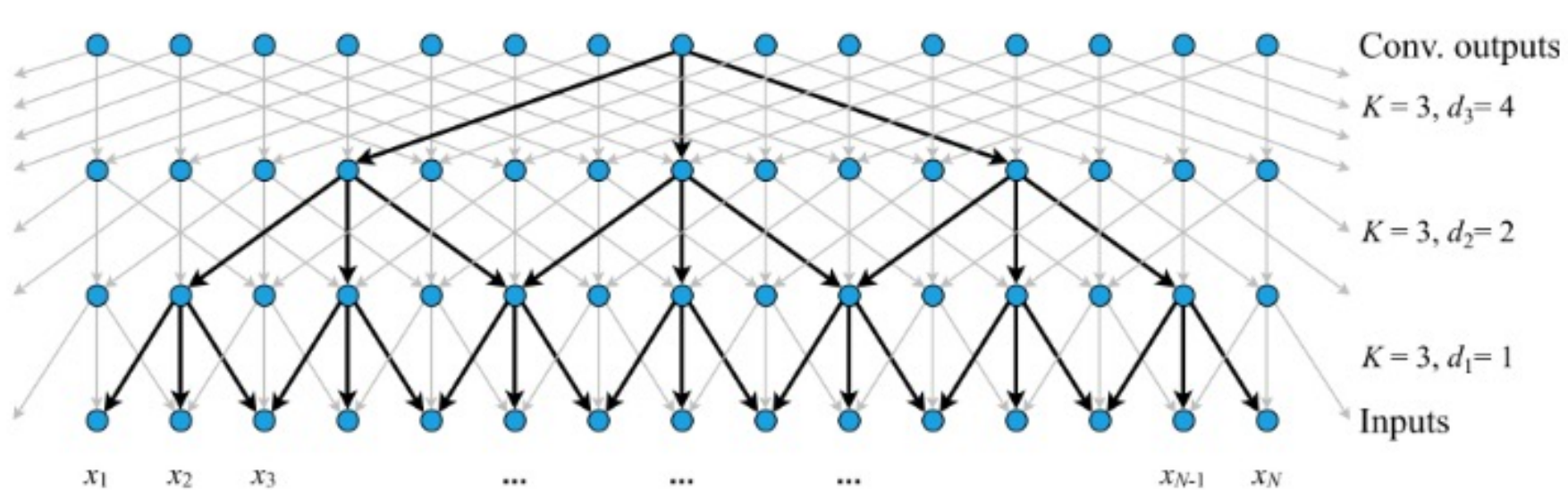
блок внимания
LDPC, Хэмминг, ...

блок Кодирования

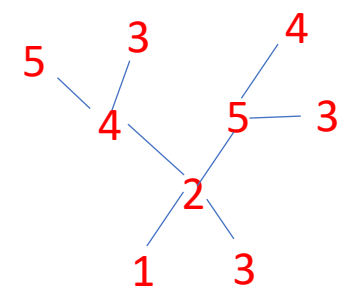
Вход

x_1 x_2 ... x_t

LDPC код может задавать структуры связей расширенных сверточных сетей dilated CNN



За глубину p построить систему связей на основе проверочной матрицы



$$Hx = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$