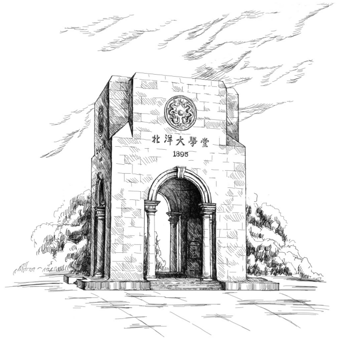
天津大学 Tianjin University
应用数学中心 Center for Applied Mathematics

NEURAL INFORMATION PROCESSING SYSTEMS

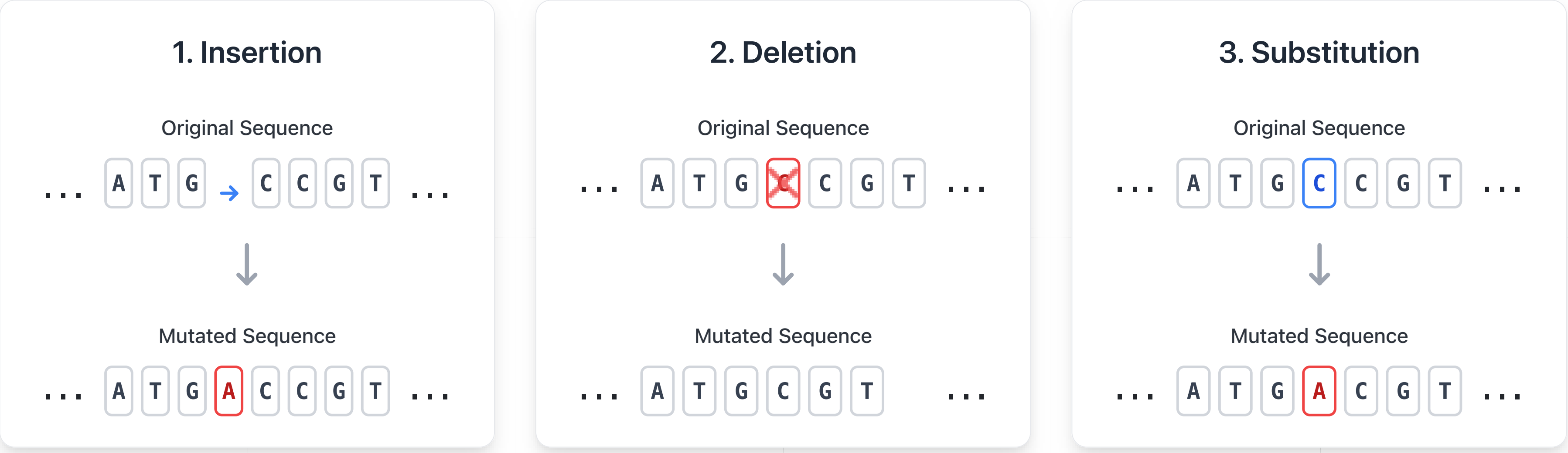
DoDo-Code: an Efficient Levenshtein Distance Embedding-based Code for 4-ary IDS Channel

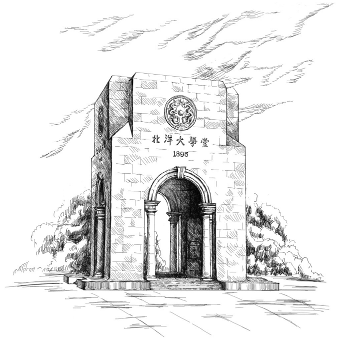
Alan J.X. Guo (郭嘉祥), Sihan Sun (孙思寒), Xiang Wei (魏祥), Mengyi Wei (魏梦怡), Xin Chen (陈鑫)



Background

Error-correcting codes for insertion, deletion, and substitution (IDS) are vital to bio-sequence information storage pipelines.





NEURAL INFORMATION PROCESSING SYSTEMS

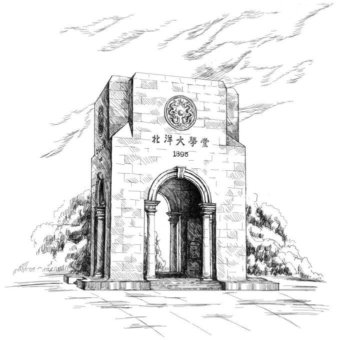
天津大学 Tianjin University 应用数学中心 Center for Applied Mathematics

Background

NEURAL INFORMATION PROCESSING SYSTEMS

Defination: [Levenshtein Distance: Distance for IDS Operations]

- The Levenshtein distance between $\boldsymbol{s}$ and $\boldsymbol{t}$ is the minimum number of IDS operations required to transform string $\boldsymbol{s}$ into string $\boldsymbol{t}$.



[1] D. Bar-Lev, T. Etzion, and E. Yaakobi, "On the size of balls and anticodes of small diameter under the fixed-length levenshtein metric," IEEE TIT, 2023.
[2] G. Wang and Q. Wang, "On the size distribution of Levenshtein balls with radius one," arXiv preprint arXiv:2204.02201, 2022.

天津大學 Tianjin University 应用数学中心 Center for Applied Mathematics

Background

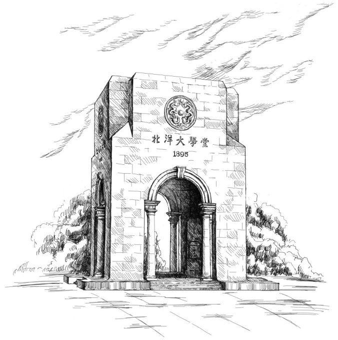
NEURAL INFORMATION PROCESSING SYSTEMS

Defination: [Levenshtein Distance: Distance for IDS Operations]

- The Levenshtein distance between $\boldsymbol{s}$ and $\boldsymbol{t}$ is the minimum number of IDS operations required to transform string $\boldsymbol{s}$ into string $\boldsymbol{t}$.

Challenge: Characterizing IDS errors remains challenging due to open problems with the Levenshtein distance [1,2]:

- High computational complexity of the metric.
- The unknown structure of the "Levenshtein ball."



[1] D. Bar-Lev, T. Etzion, and E. Yaakobi, "On the size of balls and anticodes of small diameter under the fixed-length levenshtein metric," IEEE TIT, 2023.

[2] G. Wang and Q. Wang, "On the size distribution of Levenshtein balls with radius one," arXiv preprint arXiv:2204.02201, 2022.

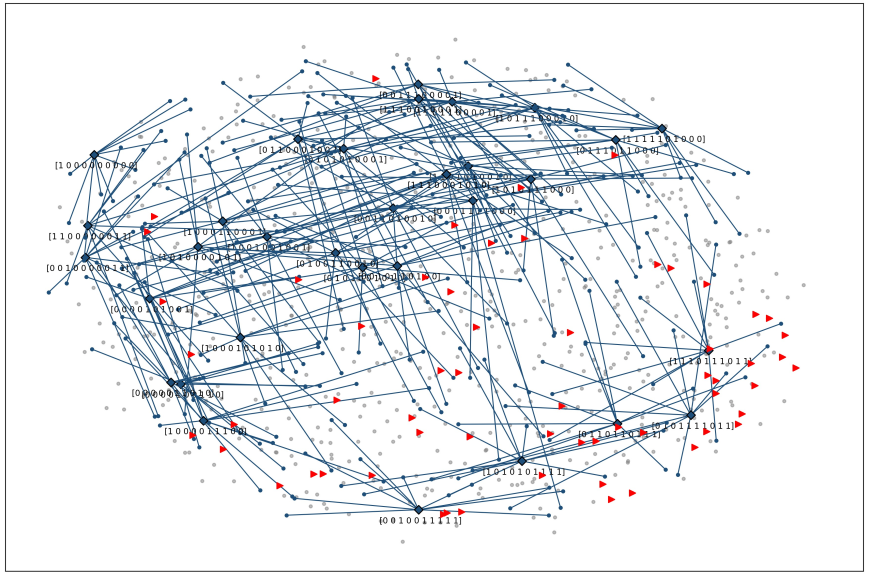
天津大学 Tianjin University 应用数学中心 Center for Applied Mathematics

Background

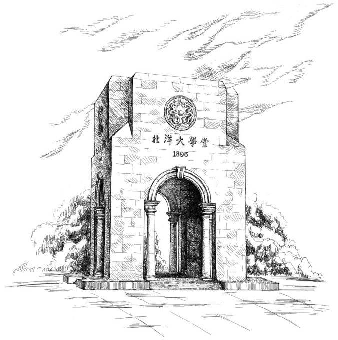
State-of-the-art 4-ary IDS-correcting codes are order optimal, requiring a redundancy of:

$$\log n + \log\log n + 7 + o(1)$$

bits [3]. This redundancy is **suboptimal** for small values of n.



Visualization of VT Code in Embedding Space.
Red marks are untapped sequences.



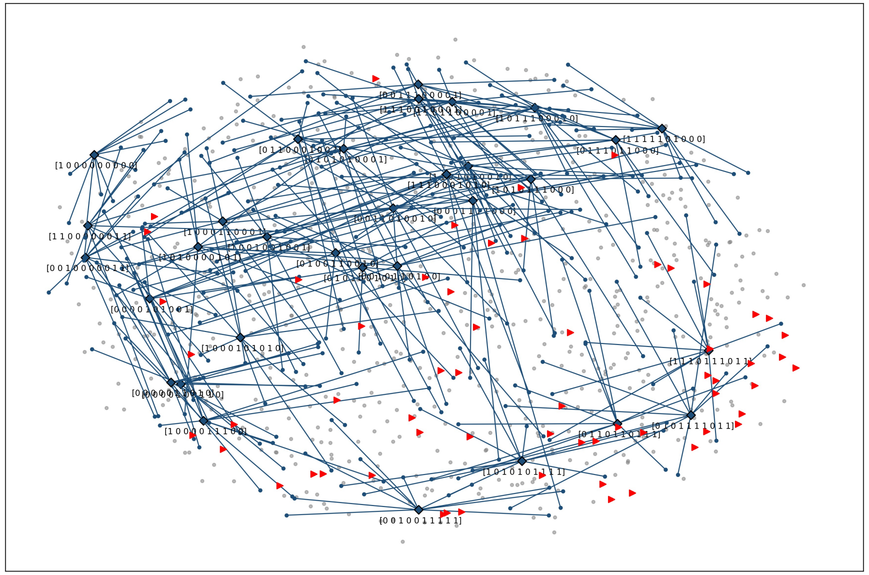
[3] Gabrys, Ryan, et al. "Beyond single-deletion correcting codes: Substitutions and transpositions." IEEE TIT, 2002.

Background

State-of-the-art 4-ary IDS-correcting codes are order optimal, requiring a redundancy of:

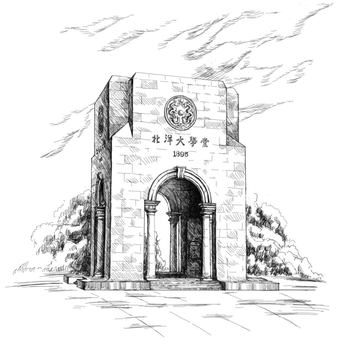
$$\log n + \log\log n + 7 + o(1)$$

bits [3]. This redundancy is **suboptimal** for small values of n.



Visualization of VT Code in Embedding Space.
Red marks are untapped sequences.

There is room for improvement in coderate. Unfortunately, we are not experts in designing combinatorial codes.

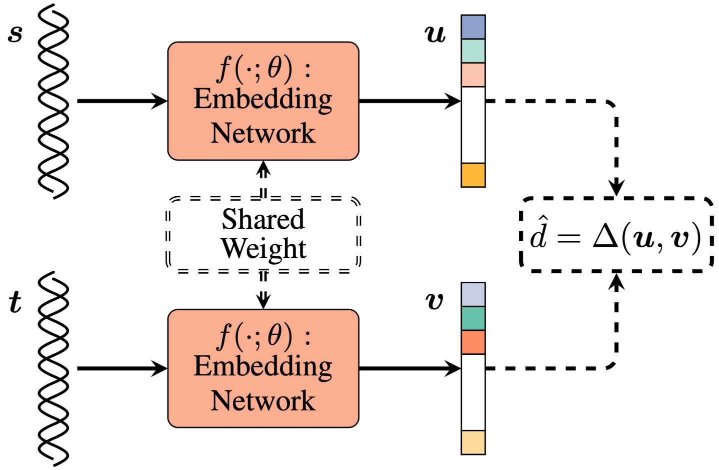


[3] Gabrys, Ryan, et al. "Beyond single-deletion correcting codes: Substitutions and transpositions." IEEE TIT, 2002.

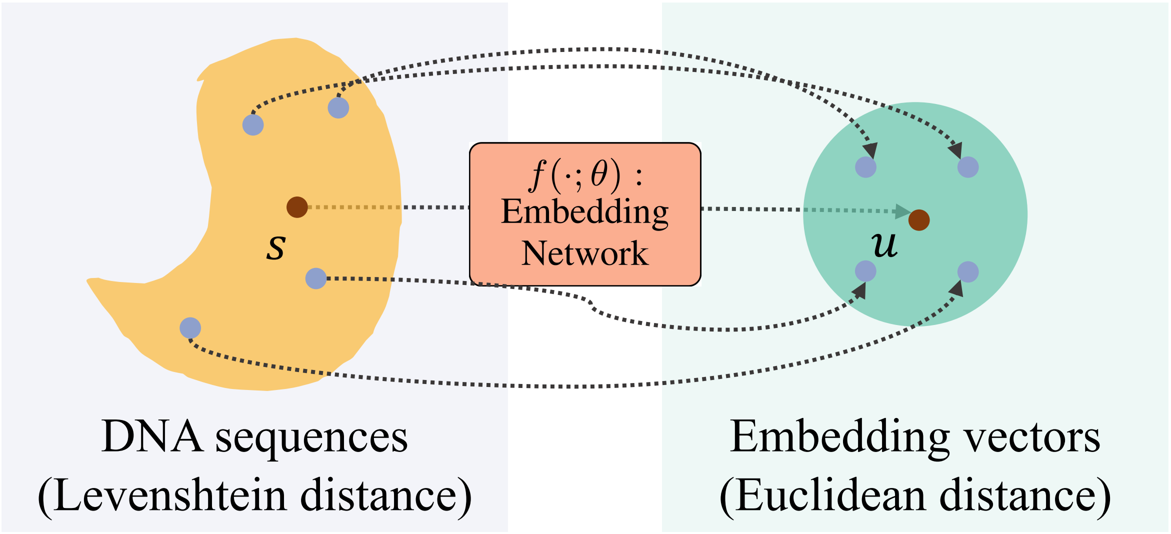
Idea

NEURAL INFORMATION PROCESSING SYSTEMS

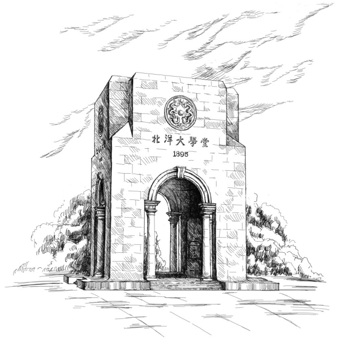
Leverage a deep embedding space[4] as a geometric proxy for the Levenshtein domain to guide code design.



Siamese neural network



Embedding network maps sequences to embedding vectors

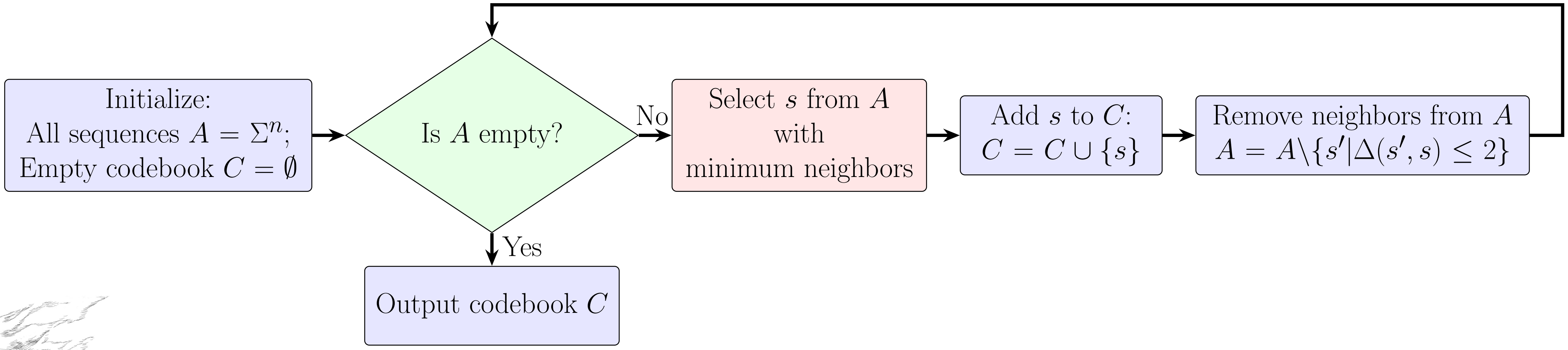


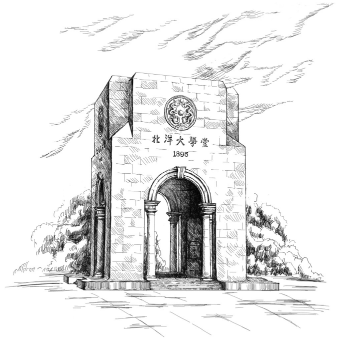
[4] Wei, Xiang, et al. “Levenshtein distance embedding with poisson regression for DNA storage.” AAAI 2024.

Method – Codebook design

Proposition: Generate a codebook C where the minimum Levenshtein distance between any two distinct codewords c_1, $c_2 \in C$ is at least 3. This codebook C can correct a single IDS error.

Greedy search of the codebook:

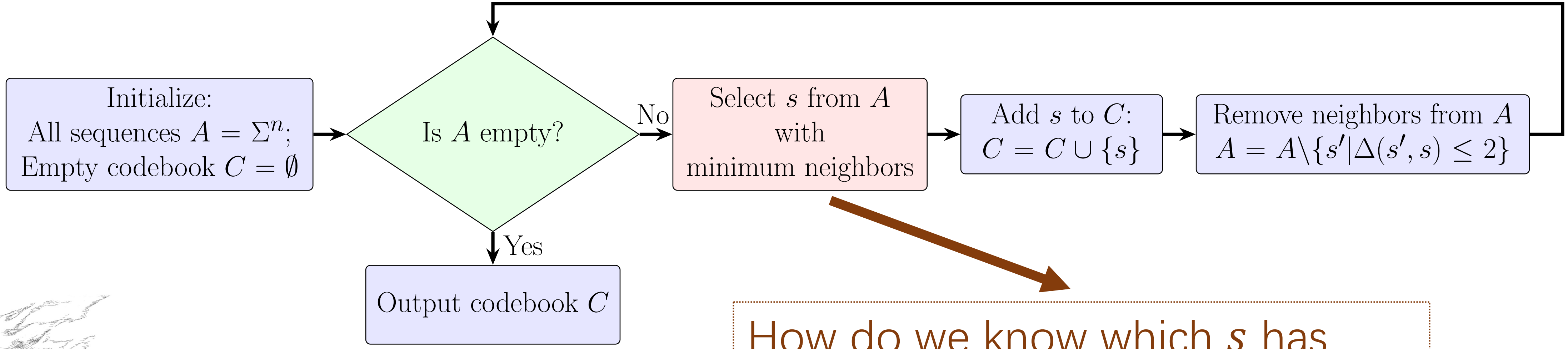




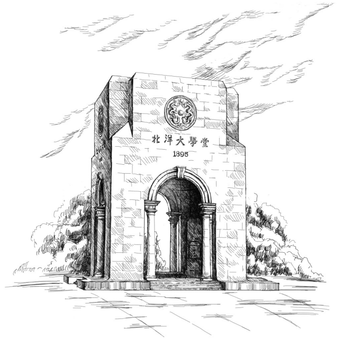
Method – Codebook design

Proposition: Generate a codebook C where the minimum Levenshtein distance between any two distinct codewords $c_1, c_2 \in C$ is at least 3. This codebook C can correct a single IDS error.

Greedy search of the codebook:



How do we know which s has minimum neighbors from A?



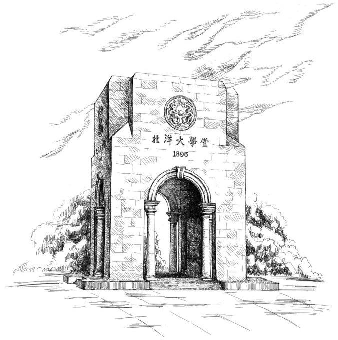
NEURAL INFORMATION PROCESSING SYSTEMS

天津大学 Tianjin University 应用数学中心 Center for Applied Mathematics

Method – Codebook design

NEURAL INFORMATION PROCESSING SYSTEMS

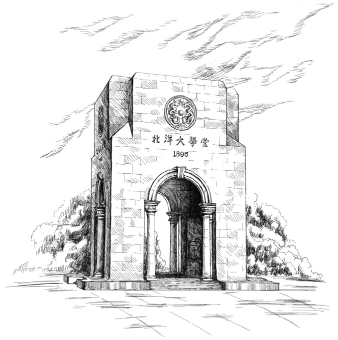
How do we know which s has minimum neighbors from A?



Method – Codebook design

How do we know which s has minimum neighbors from A?

Hypothesis:
The geometric density of embedding vectors serves as a **proxy** for the combinatorial density of neighbors in the Levenshtein domain.



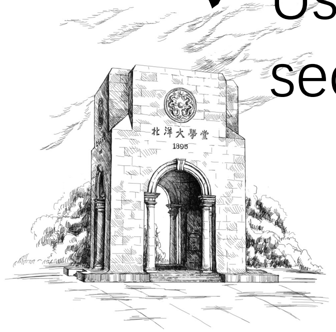
Method – Codebook design

How do we know which s has minimum neighbors from A?

Hypothesis:
The geometric density of embedding vectors serves as a **proxy** for the combinatorial density of neighbors in the Levenshtein domain.

Hypothesis:
The embedding vectors follow a multivariate normal distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$, as long as a normalization layer is applied to the embedding model.

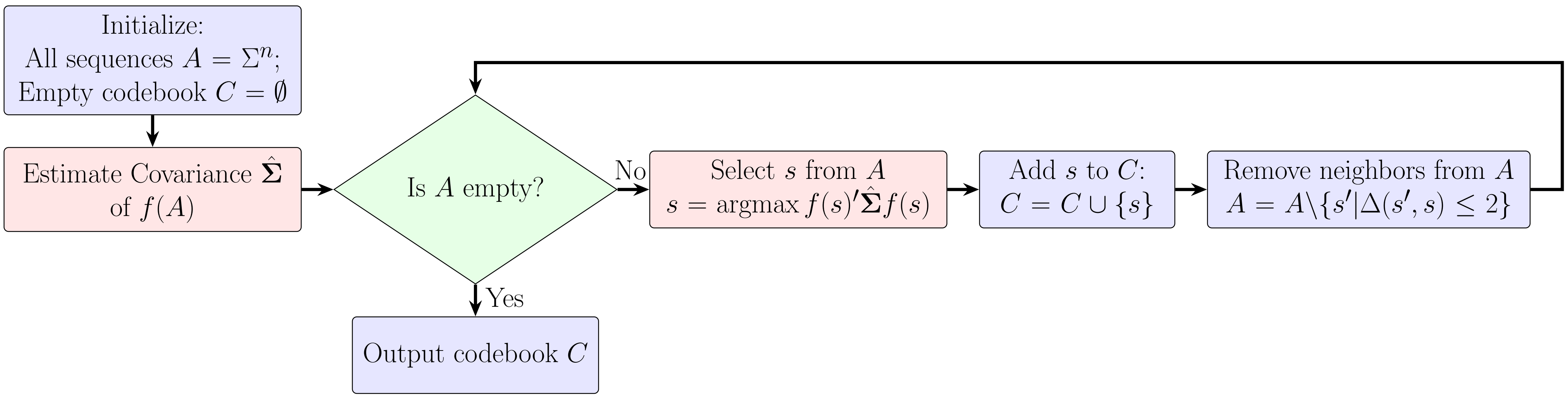
✓ Use the estimated PDF of embedding vectors as a criterion to identify sequences with fewer neighbors in Levenshtein distance.

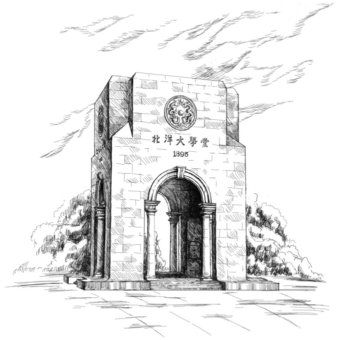


Method – Codebook design

NEURAL INFORMATION PROCESSING SYSTEMS

Implementable greedy search of the codebook:

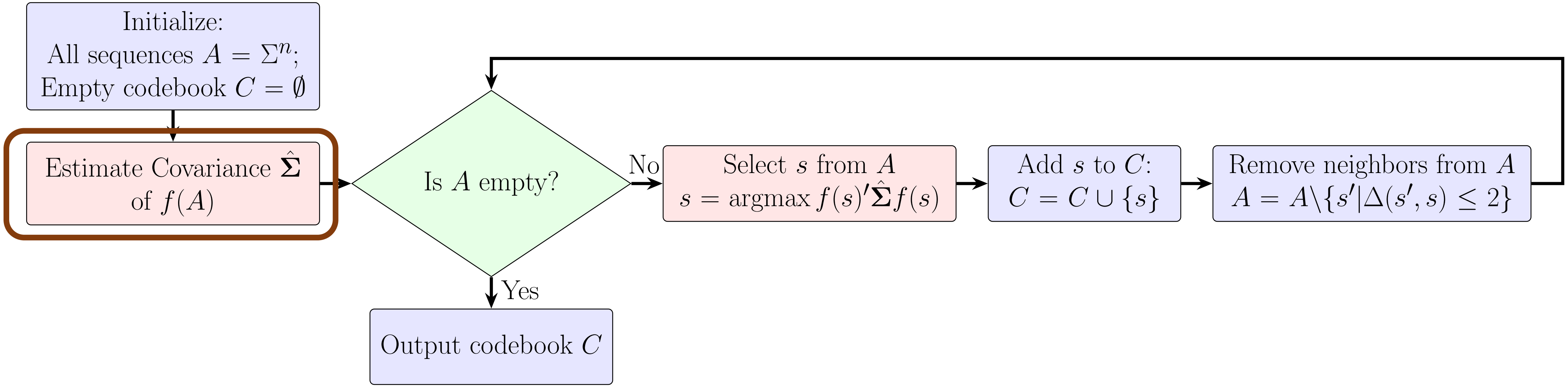


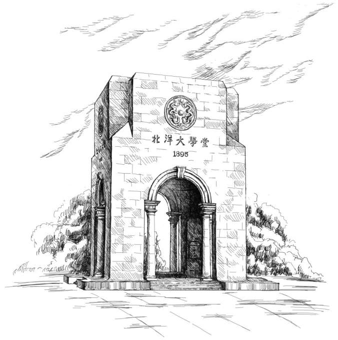


天津大学 Tianjin University 应用数学中心 Center for Applied Mathematics

Method – Codebook design

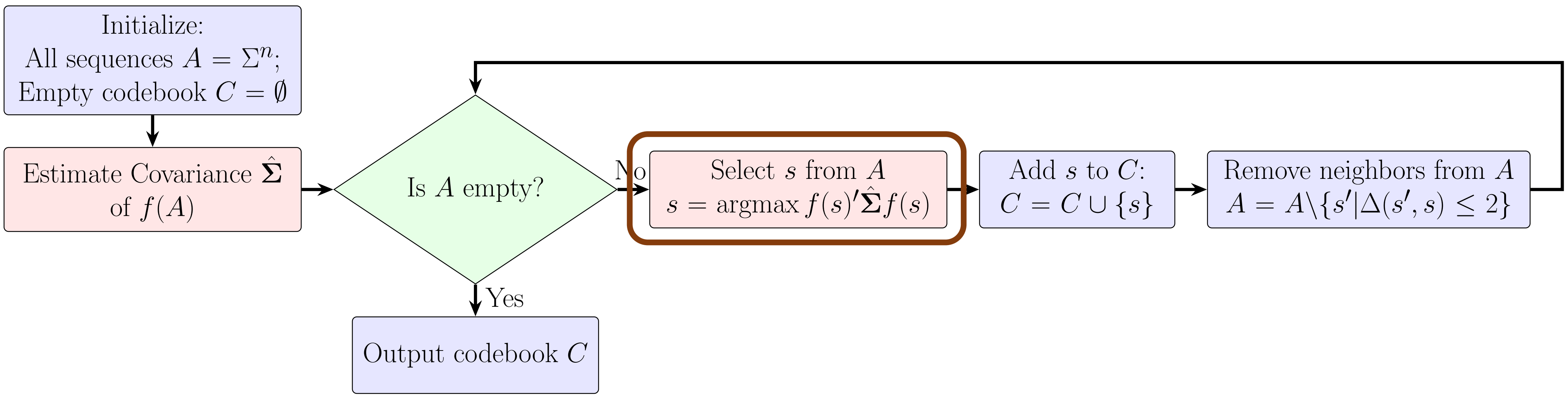
Implementable greedy search of the codebook:

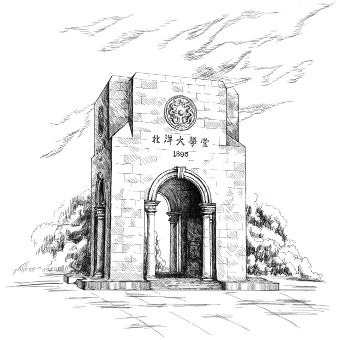




Method – Codebook design

Implementable greedy search of the codebook:





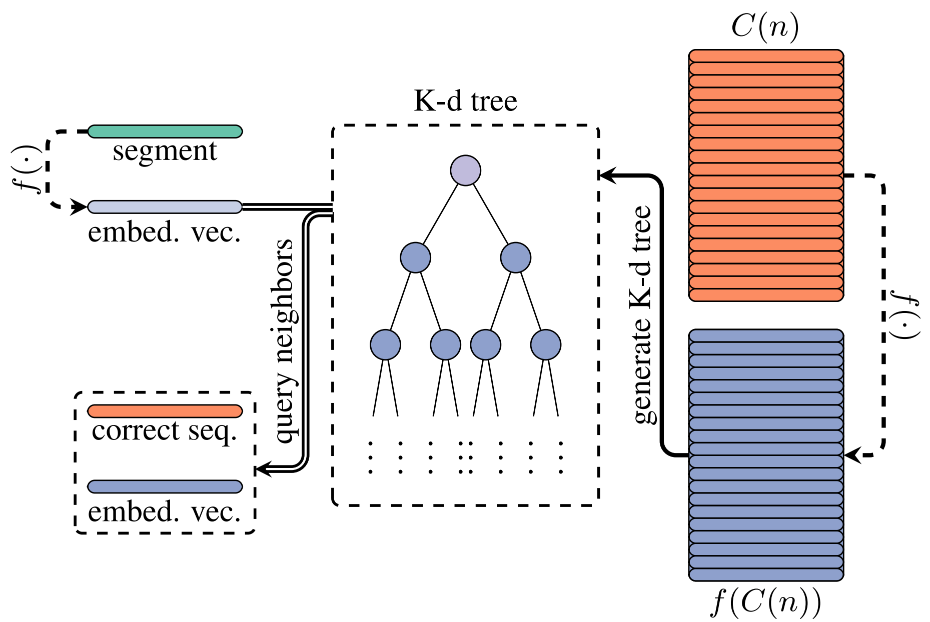
Method – Decoding in embedding space

Challenge:

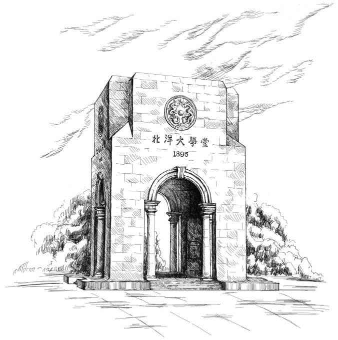
Decoding (error correcting) for this codebook is complex, driven by the high time complexity of calculating Levenshtein distance.

✓ Perform decoding in the embedding space using Euclidean distance.

✓ Mature neighbor searching methods can be leveraged.

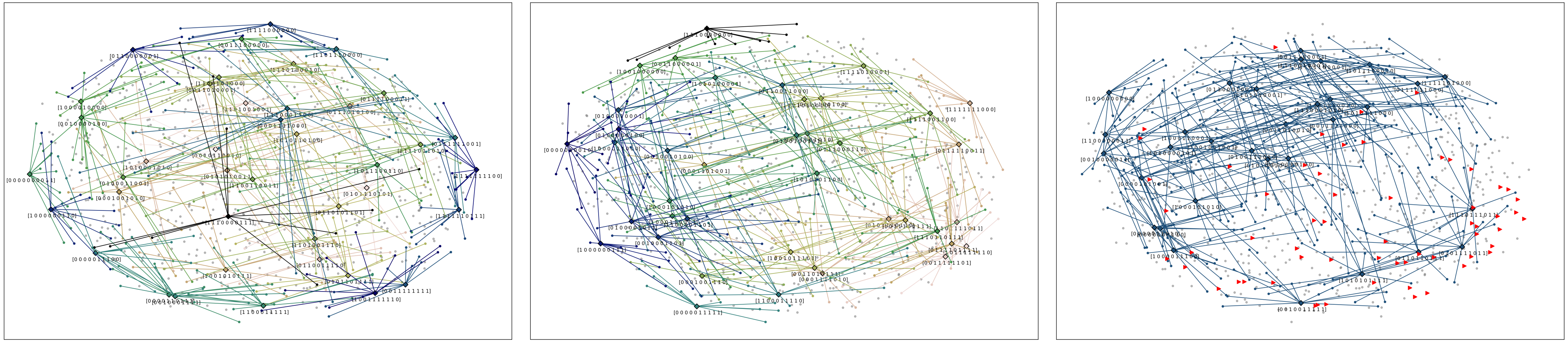


Leverage a K-d tree in the embedding space for neighbor searching (error correcting).



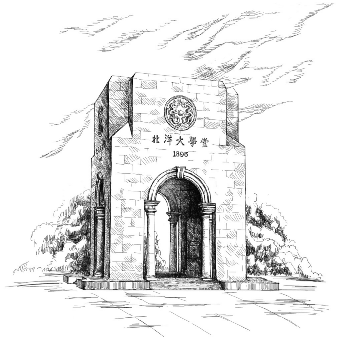
Visualization of codeword selection in embedding space

NEURAL INFORMATION PROCESSING SYSTEMS



(a) deep embedding-based search (b) random search (c) VT code

Darker colors indicate codewords selected earlier. (a) Dodo-Code: tends to select codewords from the periphery of the embedding distribution. (b) Random Search: selects codewords without any discernible pattern. (c) VT Code: leaves untapped potential codewords within the distribution.



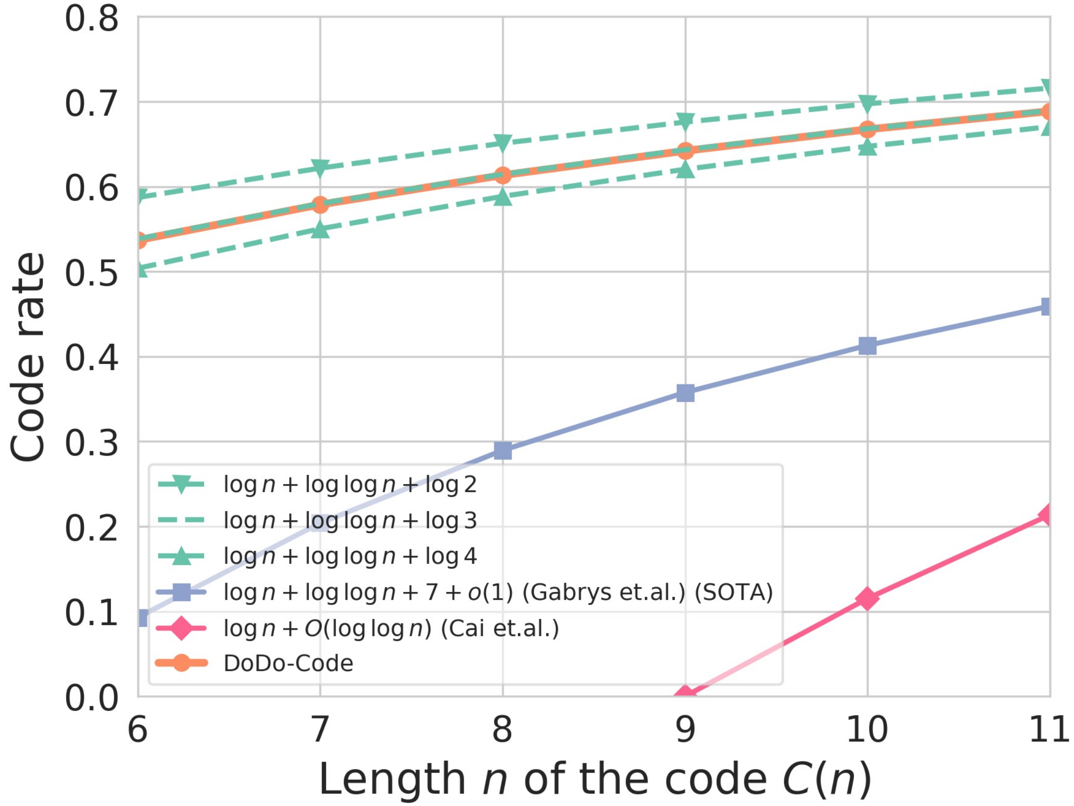
天津大学 Tianjin University 应用数学中心 Center for Applied Mathematics

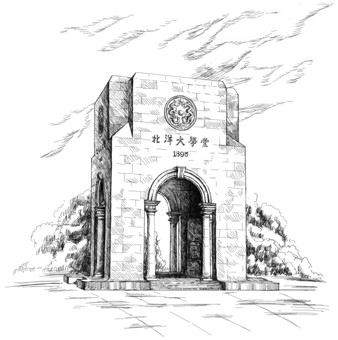
Outperforms SOTA and aligns with optimal coderate

- ✓ DoDo-Code outperforms SOTA combinatorial codes [3, 5] in coderate, when n is small.
- ✓ DoDo-Code aligns with order optimal coderate

$$\log 3n + \log\log n.$$

Note: Correcting a single substitution requires at least $\log 3n$ redundancy bits.





[3] Gabrys, Ryan, et al. "Beyond single-deletion correcting codes: Substitutions and transpositions." IEEE TIT, 2002.

[5] Cai, Kui, et al. "Correcting a single indel/edit for DNA-based data storage: Linear-time encoders and order-optimality." IEEE TIT, (2021).

天津大学 Tianjin University 应用数学中心 Center for Applied Mathematics

NEURAL INFORMATION PROCESSING SYSTEMS

Thanks for listening!

天津大学 Tianjin University

应用数学中心 Center for Applied Mathematics