
# THÈSE

**En vue de l'obtention du**

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :** *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

---

**Présentée et soutenue le** *12/12/2022* **par :**

**Lina del Pilar Bonilla Camelo**

## Theoretical framework for Time-To-First-Spike coding in Spiking Neural Networks

---

**JURY**

| | | |
|---|---|---|
| Laurie Calvet | Researcher (CR-CNRS) | Rapporteure |
| Laurent U Perrinet | Directeur de Recherche | Rapporteur |
| Jacques Gautrais | Researcher (CR-CNRS-HDR) | Co-encadrant de thése |
| Timothée Masquelier | Directeur de Recherche(DR2) | Directeur de Thèse |
| Simon Thorpe | Directeur de Recherche (DRCE2) | Invité |

---

**École doctorale et spécialité :**
*MITT : Image, Information, Hypermédia*
**Unité de Recherche :**
*Centre de Recherche Cerveau & Cognition- CerCo*
**Directeur(s) de Thèse :**
*Timothée Masquelier*
**Rapporteurs :**
*Laurie Calvet* et *Laurent U Perrinet*

# Un modèle mathématique unificateur pour le codage par temps du premier spike

Latence code

**Lina del Pilar Bonilla Camelo**

## Résumé

Au sein des réseaux de neurones impulsionnels, le codage “Time-To-First-Spike” (TTFS) utilise la latence du premier spike pour véhiculer de l’information. De nombreuses études ont démontré que ce type de codage peut transmettre beaucoup d’information rapidement. En effet, seul un spike par neurone est requis, ce qui permet d’encoder rapidement de l’information en exploitant la précision temporelle de ce spike, tout en consommant peu d’énergie.

Dans ce travail de recherche théorique, nous présentons un nouveau cadre mathématique unificateur qui permet la comparaison rigoureuse d’un grand nombre de codes TTFS existants. Nous proposons également un nouveau code, integer-based Ranked NoM, et démontrons que son pouvoir discriminatif est supérieur à celui de tous les autres codes proposés jusqu’ici.

Dans une première proposition appelée rank-order coding (ROC), les neurones sont activés au maximum quand les spikes arrivent dans l’ordre des poids synaptiques décroissants, grâce à un mécanisme de “shunting inhibition” qui désensibilise progressivement le neurone à mesure que les spikes arrivent. Dans une autre proposition appelée “$N$-of-$M$” coding, seuls les $N$ premiers spikes de $M$ neurones sont propagés, et ces motifs de premiers spikes sont déchiffrés par des neurones en aval qui utilisent des poids homogènes et pas de désensibilisation. En conséquence, l’ordre parmi les premiers spike n’importe pas.

Nous proposons un nouveau codage, “Ranked NoM” (R-NoM), qui combine des caractéristiques de ROC et NoM: seuls les $N$ premiers spikes sont propagés, mais leur ordre est déchiffré par les neurones en aval grâce à des poids inhomogènes et une désensibilisation linéaire.

Le cadre mathématique unificateur permet de comparer ces trois codes en terme de “discriminabilité”, qui représente dans quel mesure un neurone répond plus fortement à son motif préféré qu’à d’autres motifs aléatoires. Cette discriminabilité est bien supérieure pour R-NoM que pour les autres codes, et ce particulièrement dans la première phase des réponses. Nous argumentons aussi que R-NoM est bien plus adapté aux accélérateurs matériels que le code ROC original, bien que pas autant que NoM, qui n’utilise que des synapses binaires.

# An unifying mathematical framework for Time-To-first-spike coding schemes

Latency code

**Lina del Pilar Bonilla Camelo**

## Abstract

Time-To-First-Spike (TTFS) coding in spiking neural networks ( SNNs) uses the timing of the first spike to encode information. Experimental evidence indicates that a coding scheme based on the latency of the first spike transmits a large amount of information rapidly. The advantage of TTFS coding is expected because it uses precise timing and at most one spike per neuron which enables a super-fast transmission speed and low power processing.

In this theoretical research work, we present a new unifying mathematical framework, which allows rigorous theoretical comparison of a broad range TTFS coding schemes. We also present a new TTFS code, integer-based Ranked-NoM, and demonstrate that it outperforms previous proposals in terms of discriminative power.

In an early proposal, called rank-order coding (ROC), neurons are maximally activated when inputs arrive in the order of their synaptic weights, thanks to a shunting inhibition mechanism that progressively desensitizes the neurons as spikes arrive. In another proposal, called $N$-of-$M$ coding, only the first $N$ spikes of $M$ input neurons are propagated, and these “first spike patterns” can be readout by downstream neurons with homogeneous weights and no desensitization: as a result, the exact order between the first spikes does not matter.

We introduces a new potential coding - "Ranked-NoM" (R-NoM), which combines features from both ROC and NoM coding schemes: only the first $N$ input spikes are propagated, but their order is readout by downstream neurons thanks to inhomogeneous weights and linear desensitization.

The unifying mathematical framework allows to comparing the three codes in terms of discriminability, which measures to what extent a neuron responds more strongly to its preferred input spike pattern than to random patterns. This discriminability turns out to be much higher for R-NoM than for the other codes, especially in the early phase of the responses. We also argue that R-NoM is much more hardware-friendly than the original ROC proposal, although NoM remains the easiest to implement in hardware because it only requires binary synapses.

# Acknowledgements

I want to thank...

# Contents

# Scientific paper

Bonilla, Lina and Gautrais, Jacques and Thorpe, Simon and Masquelier, Timothée.
**ANALIZING TIME-TO-FIRST-SPIKE CODING SCHEMES: A THEORETICAL APPROACH**.
ORIGINAL RESEARCH article *Front. Neurosci., 2022 Sec. Neuromorphic Engineering*

https://www.frontiersin.org/articles/10.3389/fnins.2022.971937/full

# List of Figures

# List of Tables

# 1 Introduction

Modern computing systems consume far too much energy. We are so accustomed to accessing information near-instantaneously that we neglect the energy –and therefore environmental–consequences of the computing systems giving us this access. Nevertheless, each Google search has a cost: data centres currently use around $200$ terawatt hours of energy per year, forecast to grow by around an order of magnitude by $2030$ [30]. Guided by brain-like "spiking" computational frameworks, neuromorphic computing promises to realize artificial intelligence while reducing the energy requirements of computing platforms [43].

The human brain performs impressive feats, for example, simultaneous reasoning, recognition, control and movement with a power budget of about $20$W (watts)[6]. By contrast, a standard computer performs only recognition among $1000$ different kinds of objects expends about $250$W [43]. This is largely due to its architecture which consists of dense networks of neurons transmitting signal through their synapses efficiently. Inspired by the brain's hierarchical structure and neuro-synaptic framework, state-of-the-art AI is implemented using artificial neural networks (ANN).

## 1.1 Neural Networks

Neural dynamics can be conceived as a summation process(sometimes also called "integration" process) combined with a mechanism that triggers action potentials(spikes) above some critical voltage.

Maass [25] categorizes neural networks into three generations based on their underlying neuronal funcionality. The first generation, referred to as McCulloch-Pitt perceptrons. Modern deep-learning networks (DLNs) belong to the second generation which implemented continuous non-linearity to the neural unit (e.g. sigmoid unit or rectified linear unit), therefore, their continuous neuronal functionality of these models supports gradient-descent-based back-propagation learning, the standard algorithm for training DLNs today.

The third generation of networks uses spiking neurons that exchange information via spikes. Spikes are essentially binary events, either $0$ or $1$. The most important distinction between these network generations is in THE NATURE OF INFORMATION PROCESSING. The second generation uses real-valued computation, that is, floating-point variables to represent both the neuronal activation levels and the strength of synaptic connections, whereas SNNs can uses the timing of the spikes to process information. A neuronal unit in a SNN is only active when it receives or emits spikes, which can contribute to energy efficiency over a given period of time. In contrast, in DLNs all units are active regardless of the real-valued input or output value. Furthermore, the fact that the inputs in an SNN are either $1$ or $0$ reduces the mathematical dot-product operation-the integration-to a less computationally intensive summation operation.

### 1.1.1 Neural Coding

Neurons represent and transmit information by firing sequences of spikes (spike train) in various temporal patterns. Static-frame-based data on SNNs are converted to spikes trains using appropriate encoding techniques such as temporal coding or rate coding. Spike timing and firing rate are measures to quantify neural spike train data [14].

#### Spike timing code and rate code

Spikes can be characterized as discrete events, with a relatively well-defined time, which we can identify to the time of release of synaptic vesicles at the axon terminal. Thus a single spike is associated with physically measurable quantities, and it forms the basis of communication between neurons on a fast timescale. To a large extent it can be characterized by its timing [4].

Firing rate code is an abstract concept that is defined in a limit that involves an infinite number of spikes. A quick glance at the experimental literature reveals that there is no unique and well-defined concept of firing rate. In fact, there are at least three different notions of rate, which are often confused and used simultaneously [14]. The three definitions refer to three different averaging procedures: over time (number of spikes divided by the duration, in the limit of infinite duration), over neurons (average number of spikes in a population of neurons, in the limit of an infinite number of neurons) or over trials (average number of spikes over an infinite number of trials) [4].

*Thus these are two different concepts: spike timing is what defines spikes trains, whereas rate is an abstract mathematical construction on spike train.*

These measures are useful tools to study a neural system but the question is whether neurons transmit information by using any of theses quantities as a neural code.

#### Why firing rate code is not a good idea?

Many researchers assume that neurons send information using a firing rate code in which the neuron's activation level is represented by the number of spikes emitted in a given time window. If that was the case, replacing the firing rate with a floating-point number is a perfectly reasonable strategy. However, it has been argued that this sort of firing rate code would be intrinsically very inefficient because you would need a lot of spikes to encode

information with any degree of accuracy [12]. For example, suppose that we wanted to represent the activation level with a precision of $8$-bits. To do this using a conventional rate code would mean waiting long enough for the neuron to emit $255$ spikes when maximally activated – and this would mean waiting for a second or more to make even the most basic decisions. This very low efficiency has led some researchers to rule out spike-based coding schemes. They point out that it is much simpler, and much more accurate, to represent information as a floating-point number that can be transmitted in a single clock cycle via a $32$-bit bus.

You could argue that there are alternative ways of implementing a firing rate based code that are much faster. For example, rather than sending an $8$-bit activation level using a single neuron that emits between $0$ and $255$ spikes in a given time window, you could have $255$ neurons in parallel, each of which only needs to emit at most one spike in, say, $10$ milliseconds. But this sort of population rate coding scheme would also be very inefficient because it would need very large numbers of neurons.

You might also argue that it is possible to estimate the instantaneous firing rate of a neuron by looking at the interval between two spikes. An interspike interval of exactly $4.0$ ms would correspond to an instantaneous firing rate of $250$ spikes/second. And, in such a case, the accuracy with which the underlying rate can be determined would be limited only by the temporal precision with which the neuron can emit spikes. If the precision was $0.1$ ms, you could encode many different activation values in $25$ ms. But while possible in principle, such a scheme would require very complex mechanisms to decode as well as being unusable until the neuron has emitted $2$ spikes.

It would appear that the fundamental problem here is that researchers have apparently been assuming that spike-based coding has to be some sort of rate coding scheme. But this is certainly not the case.

### 1.1.2 Encode information

### 1.1.3 Intensity to delay converter

Even the simplest neuronal models have the property that the time taken for a neuron to reach threshold depends on the intensity of the input. And this means that the latency of the first spike in response to a stimulus can be used as a code. Remarkably, variations in spike latency with input intensity were demonstrated in the very first recordings of activity in the optic nerve by Lord Edgar Adrian in Cambridge in the 1920s [2]. But this basic physiological fact was essentially ignored for several decades, before being demonstrated again by neurophysiological studies[15].

Therefore, timing of the first spike provides an alternative way to encode information – a scheme known as time-to-first spike coding (TTFS) –, there are a number of very interesting options that can be considered.

Historically, TTFS was first proposed to explain the phenomenal speed of processing in the brain for certain tasks, such as object recognition [50]. More recently, TTFS has attracted much attention from the AI community [32, 44, 58, 21, 37, 46, 57, 5, 31], because it can be efficiently implemented on low power event-driven neuromorphic chips [1, 33, 35, 47, 16, 24], leveraging two key features.

### 1.1.4 Spike timing in a sparse sensory code

The first one is sparsity [9]. Neurons fire at most once, but usually most neurons do not fire at all. Processing thus consumes very few spikes, and thus very little energy, because usually idle neurons do not consume much [7]. The second one is time. If using event-driven processing, for example, address event representation (AER), time represents itself [29]. Thus one can compute with time without ever storing timestamps. For example, a decision can be made based on the first neuron to fire in the readout layer. And this is possible even if the firing time difference is infinitesimally small. Conversely, a readout based on the activation levels requires storing these activation levels with high precision to be able to always distinguish the most active neuron.

Gupta et al. [18] show that even remarkably sparse spiking responses can provide information through stimulus-specific variations in timing on the order of tens to hundreds of milliseconds and that these variations can determine the responses of downstream neurons. These results establish the importance of spike timing in a sparse sensory code.

To code information efficiently, sensory systems use sparse representations. In a sparse code, a specific stimulus activates only few spikes in a small number of neurons. A new study shows that the temporal pattern across sparsely activated neurons encodes information, suggesting that the sparse code extends into the time domain. [22]

## 1.2 Time-To-First-Spike: latency code

Time to first spike (TTFS) coding scheme transmits information to the destination neurons on the arrival of the first spike, which enables a super-fast transmission speed. Many experiments have pointed out the significance of the first spikes in various parts of our nervous system, such as retina, auditory systems, and tactile afferents [40]. Various works have reported that applying TTFS coding scheme in SNNs can significantly reduce the number of spikes and improve inference speed [45]; Oh et al., 2020; [38]).

Experimental evidences indicate that a coding scheme based on the latency of the first spike transmit a large amount of information. For example touch sensors in the finger tip encode the strength and direction of the touch in the timing of the first spike emitted by each neuron[14] Similarly, the relative latency of first spikes of retinal neurons encode the image projected on the retina [14]. Thorpe et al. [52] argues that the brain does not have time to evaluate more than one spike from each neuron per processing step. Therefore the first spike should contain most of the relevant information, which is read out by neurons further down the processing chain. Using information-theoretic measures on their experimental data, several groups have shown that most of the information about a new stimulus is indeed conveyed during the first 20 or 50 milliseconds after the onset of the neuronal response [53],[41].

Guo et al.[17] performed an comparative study on the impact and performance of different neural coding The results show that TTFS coding is the best choice in achieving the highest computational performance with very low hardware implementation overhead. They also demonstrated the excellent performance of TTFS coding during inference in terms of classification performance and power consumption.

Park et al.[38] compared different neural coding schemes in terms of classification

accuracy, latency, the number of spikes, and energy during inference. The comparison revealed that TTFS coding won against the other coding schemes in classification and computational performance.

## 1.3 Two powerful TTFS codes

This work will be inspired and based on two TTFS temporal codes: firstly, *Rank order coding* [13] [49] which considers the neuron as an intensity to delay converter. Given this asynchrony in spike generation, the way they developed the coding scheme was to throw away the precise timing information, and use a code that depends only on the order in which the spikes arrive. And secondly, $N$*-of-*$M$ *coding*, where $N$ of the $M$ neurons in a bundle fire at (approximately) the same time to convey information. The information content is defined by the choice of the $N$ firing neurons.

### 1.3.1 Rank Order Coding

Rank Order Coding proposed a mechanism for decoding the order of synapse activation to all the synapses of a neuron: the first synapse activated will be with full efficiency, the second with less efficiency, etc. Neurons are maximally activated when inputs arrive in the order of their synaptic weights, thanks to a shunting inhibition mechanism that progressively desensitizes the neurons as spikes arrive. Synapses therefore were governed by two values: the synaptic weight, given or adjusted by learning, temporally stable, and the efficiency of this weight, dynamically determined by its activation rank, which depends on the stimulus. This mechanism provided a simple and efficient way to decode the order of activation of inputs.

#### Mechanism

According to the results obtained by [26] on a train of impulses arriving at a synapse, the decreasing efficiency of the weights can be modeled by a geometric sequence of factor $m$, where $0 < m < 1$. The efficiency thus follows a decreasing exponential. The weight actually transmitted will therefore be the original weight multiplied by the modulation factor to its rank:

$$w_{i,\text{effective}} = w_i m^{\text{Rank}(w_i)-1} \tag{1.3.1}$$

After k activated synapses, the activity propagated by these synapses is given by the sum of these weights affected by their modulation according to their rank: [13]

$$S^{(k)} = \sum_{i=1}^{k} w_i m^{(i-1)} \tag{1.3.2}$$

The final value of this estimator is obtained for $k = n$ weights that can be activated. The weights form a permutation of the first n integers.

Equivalently, at the paper [49] the activation level of neuron $i$ at time $t$ is given by

$$\text{Activation}(i,t) = \sum_{j \in (1,M)} \text{mod}^{\text{order}(a_j)} W_{j,i} \tag{1.3.3}$$

where order$(a_j)$ is the firing rank of neuron $a_j$ in the ensemble of afferent neurons $A = a_1, a_2, \cdots, a_m$, with $W = w_{1,j}, w_{1,j}, \cdots w_{m,j}$ the weights of the $m$ corresponding connections and mod be an arbitrary modulation factor between $0$ and $1$ .

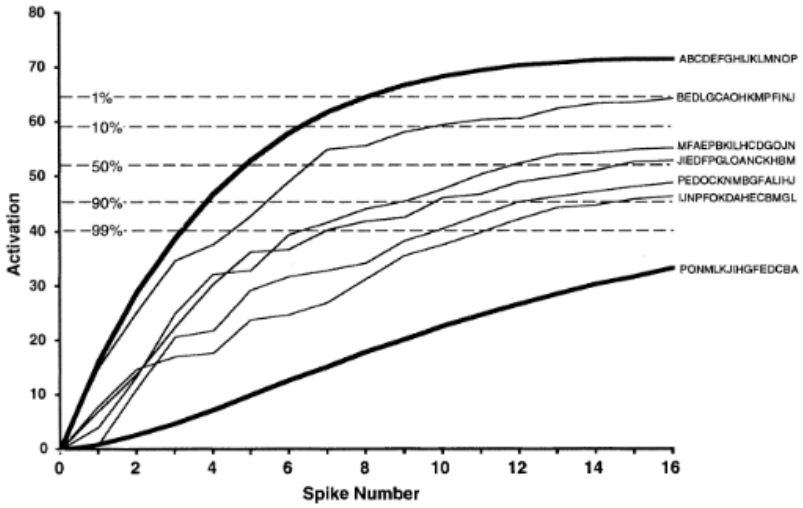



Figure 1.1 – The increase in activation level depends on the order of firing. Maximal activation occurs when the inputs fire in the order of the weights $(A, B, \cdots, P)$. Activation is minimal when the order is reversed. Intermediate lines correspond to $5$ randomly selected input patterns chosen from the $16! = 20,922,789,888,000$ possible input spike orders. The five dotted lines specify the proportion of such random patterns that will exceed a given final activation level.

### Linear Rank Statistics

The statistical approximation of the behavior of the sum of the weights modulated by their rank, (which we call integration) were based on linear rank statistics. For any statistic of the form

$$S^{(n)} = \sum_{i=1}^{n} c_i^{(n)} a^{(n)} R_i^{(n)} \tag{1.3.4}$$

is called statistics of rank linear, where $n$ quantities $a^n(i), i = 1, ...., n$ are called *scores* (these are the factors of modulation), the $c_i^{(n)}$ are often referred to as regression constants (which are the synaptic weights) and $R^n = (R_1^n, ..., R_n^n)$ a random rank vector has values in the set of the permutations of $(1, ..., n)$. The two first moments of $Sn$ are computable because they only depend on the first two moments of $a^{(n)}$ and of $c^{(n)}$:

$$E(S^{(n)}) = n\bar{c}^{(n)}\bar{a}^{(n)}, \qquad Var(S^{(n)}) = (n-1)\sigma_a^2\sigma_c^2 \tag{1.3.5}$$

with the linear weights $c^{(n)} = i$, and the modulation geometric $m$, $a^{(n)} = m^{i-1}$, the expected value and the variance were given by,

$$E(S^{(n)}) = \frac{(n+1)}{2}\frac{(1-m^n)}{(1-m)} \tag{1.3.6}$$

$$Var(S^{(n)}) = \frac{n}{(n-1)} \frac{(n^2-1)}{12} \left( \frac{1-m^{2n}}{1-m^2} - \frac{1}{n} \frac{(1-m^n)^2}{(1-m)^2} \right) \tag{1.3.7}$$

They showed the asymptotic normality of the estimator $S^{(n)}$ and with the first order moments, $E(S^{(n)})$ and $Var(S^{(n)})$, they could therefore build a statistical test on this estimator.

Thus, they concluded that the neuronal unit model performs the statistical test on a stimulus encoded by the order of the activation of the weights, and accurately described the effect of the effective threshold value on the partition of the set of stimuli.

#### Speed and Accuracy trade-off

They also proposed another interesting feature of the Rank Order decoding mechanism that by adjusting the threshold level, the output cell will fire at different times depending on how well the input pattern matches the weights.

For instance, setting the threshold at around the $50\%$ level in Figure 1.1, the neuron would fire after only five inputs have been activated if the order was perfect (i.e. $ABCDE$). With random inputs, firing when only $5$ inputs have been activated would only occur roughly $0.01\%$ of the time, and the few sequences that do work would be very close indeed to optimal. As more and more spikes arrive, the proportion of patterns capable of resulting in a spike increase, from $4.4\%$ after $6$ inputs have fired, $14\%$ after $8$ have fired and $50\%$ when all $16$ inputs have fired. Thus, varying the threshold provides a simple way of adjusting the speed-accuracy trade-off of such a mechanism [49].

### 1.3.2 $N$-of-$M$ Coding

$N$-of-$M$ situation is effectively like an order code, but where you fix the number of active input lines and ignore the order in which they fire. Effectively, you lose information by ignoring the precise ordering, but its straightforward to show that it is very efficient.

A valid $N$-of-$M$ binary code consists of exactly $N$ asserted bits selected from a total of $M$ bits. In other words, in a population of $M$ neurons exactly $N$ fire for each symbol. $N$-of-$M$ encoding combines a relatively high information capacity with intrinsic self-timing property [55]-the completeness of the data is implicit in the coding- so consecutive layers of neurons "know" when they have complete input information and can use this to trigger the generation of their outputs. In addition a $N$-of-$M$ code has the following important properties:

- They require a constant, low (in the case of sparse codes), average neural activity, resulting in good power-efficiency;
- they have strong error recovery properties due to their sparse distribution in the $M-$ dimensional binary space, giving robust operation;
- When comparing two random $N$-of-$M$ codes, one $N_1$-of-$M$ and the other $N_2$-of-$M$, the statistics of the likely match are of interest;

- The probability that two such random $N$-of-$M$ codes share $x$ 1s follows the hypergeometric distribution:

$$P(X = x) = \frac{\binom{N_2}{x}\binom{M - N_2}{N_1 - x}}{\binom{M}{N_1}} \tag{1.3.8}$$

- The number of matching 1s, $x$, can take only integers values, and so the probability of meeting a matching threshold can be controlled only coarsely by adjusting that threshold.

- the number of matching 1s, $x$, will be between $0$ and min$(N_1, N_2)$

We will perform in the Chapter VI a proof that $N$-of-$M$ codes follows the hypergeometric distribution by linear algebra and combinatorics.

It is of interest to note that, in the case of unordered $N$-of-$M$ codes, the nonnormalized scalar product is linearly related to the Hamming distance $H(x; x')$. The relationship is

$$< x, x' >= N - H(x; x')/2 \tag{1.3.9}$$

which may be explained by observing that the number of jointly active bits in $x$ and $x'$ is equal to the total number of active bits $(N)$ less the number of bits that are active in $x$ but not in $x'$ $(H/2)$ [**article**].

## 1.4 Applications TTFS codes

One of the applications of TTFS code is the sparse distributed memory (SDM) which is a form of associative memory. SDM represents an information storage device in the computer science [23] and psychology [48] literature as a plausible mathematical model of human long-term memory.

The original SDM was proposed by Kanerva [20] as a model of human long-term memory; however, since this original work, much has come to light on the coding of human memories in terms of temporal properties, i.e., precisely when a neuron fires relative to others—the rank order of the neurons [**article**], [39], [51].

### 1.4.1 Sparse Distributed Memory using $N$-of-$M$ neural codes

Furber [11] proposed an efficiency and simpler structure of a sparse distributed memory based upon the use of $N$-of-$M$ codes.

Conventional (dense) binary codes as used in Kanerva's SDM are ill-suited to SN implementations. If a spike represents a $1$ then a neuron can readily detect the 1s in the input code, but it has no simple way to detect the $0$s. This problem is avoided if the code has a fixed number of 1s, since once the 1s have been detected the $0$s are located implicitly. This leads Furber to the use of $N$-of-$M$ codes, where in a population of $M$ neurons exactly $N$ fire for each symbol. The suitability of $N$-of-$M$ codes for SN implementations is closely related to their self-timing properties (implicit) which are:

- the completeness of the data (so consecutive layers of neurons 'know' when they have complete input information and can use this to trigger the generation of their outputs)
- In most cases, the coding is sparse ($N << M$) and this result in a constant low average neural activity.

In addition $N$-of-$M$ codes have good error-recovery properties, and this makes the proposed system relatively insensitive to noise and other sources of error [**article**].

The memory has a higher efficiency and a much simpler basic structure that Kanerva's sparse memory based on binary codes. It is also well-suited to implementation using pulsed neurons, requiring only unipolar binary synaptic weights with a simple Hebbian learning algorithm and yielding a constant, low, average neural activity level.

## Memory architecture

The memory comprises two neural structures (two layer network): the address decoder (fixed layer) and the data memory (data store layer). The address decoder $A$ matrix is a $W \times M$ binary matrix, where $W$ is the number of *address decoders*. Each row (decoder) in $A$ has a fixed number of bits set $a$ chosen randomly from the total $M$.

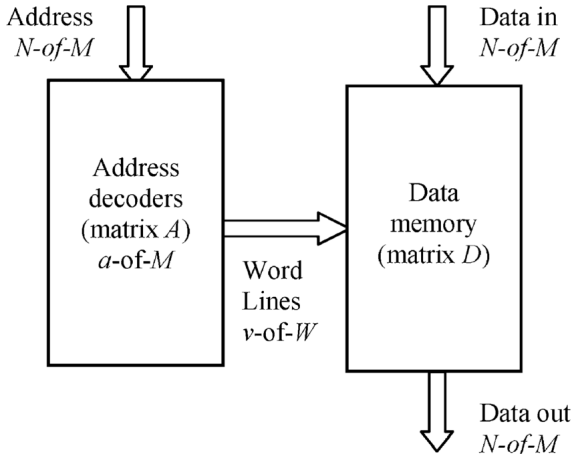



Figure 1.2 – Structure of an SDM that uses $N$-of-$M$ codes. The address decoder has $W$ outputs that each receive inputs from a randomly selected a of the $M$ input address elements. For each input address, the $v$ address decoder neurons with the highest activation fire. The data memory is a fully connected $W \times M$ correlation matrix memory. [11]

The **data store layer** is just a CMM (correlation matrix memory) with $N$-of-$M$ input and output encoding. The **address decoder** threshold $T$ can be related to the number of address decoders that fire, $w$. The firing address decoders include all those that have $T$ or more bits in common with the input address, that is, the number of times random $N$-of-$M$ codes share $x1$s. Therefore, the expected value of $w$ is given by the expectation of the hypergeometric distribution (Equation 1.3.8).

### 1.4.2 Sparse Distributed Memory using Rank-Order neural codes

An SDM based upon rank-order codes was built from the structure of the SDM as illustrated in Fig. 1.2. Rank-order codes share the same properties as unordered $N$-of-$M$ codes – intrinsic self-timing (see Section 1.4.1). An attractive feature of rank-order encoding is its potentially high information content compared with binary and unordered $N$-of-$M$ encoding. There are $N!$ times as many rank-order $N$-of-$M$ codes as there are unordered $N$-of-$M$ codes. Rank-order coded SDM can employ a structure with a very similar cost to the unordered $N$-of-$M$ SDM, and the additional information content in the rank order can be stored and recovered at very little additional cost.

Furber [10] had thus shown that a simple binary SDM can be used to store and recover rank-ordered codes with good efficiency and that the memory has appreciable error-recovery capabilities that render it usable in complex noisy systems. They showed further that human memories may be stored in terms of neural spike timings.

### 1.4.3 Implementing the memory using SNN

Furber [10], offered pointers for implementing SDM using Spike Neural networks. The possible architecture capable of detecting and generating rank-ordered $N$-of-$M$ codes is shown in Figure 1.3.

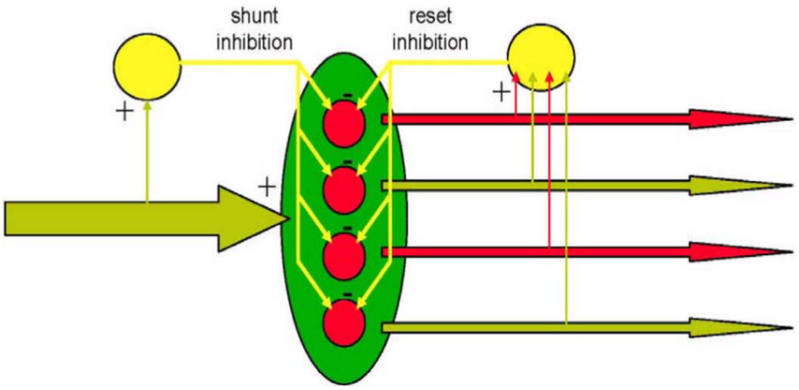



Figure 1.3 – . The feedforward shunt inhibition on the input side is used to reduce the gain of the main bank of neurons, thereby making them sensitive to the rank order of the inputs. The feedback reset inhibition on the output side is used to prevent the number of neurons that fire in each burst exceeding $N$ [**article**].

.

The shunt inhibition (see Fig. 1.3) has the effect of dividing the total excitory input by some factor, which is implemented by using multiplication for a factor less than $1$. This means that the activation $\Omega$ of an individual neuron (here, they used the leaky integrate-and-fire model) can be described in the following form:

$$\dot{\Omega} = -\Omega/\tau_{\Omega} - \sum(w_{ij}x_j)G \tag{1.4.1}$$

where $\tau_{\Omega}$ is the "leaky" time constant, $w_{ij}$ are the input connection weights, $x_j$ are the inputs (assumed here each to be a time series of delta function spikes), and $G$ is the

shunting inhibition factor, which follows a similar leaky integrating form

$$\dot{G} = (1 - G)/\tau - \alpha \sum x_j \tag{1.4.2}$$

where the resting value (in the absence of inputs) is $1$, $\tau$ is the time constant, and $\alpha$ is a scale factor. If the input burst representing a rank-order code is equally spaced over time, with a separation $t$ between consecutive input spikes, then, the $n$th input will be modulated by

$$G_n = 1 - ke^{-t/\tau}(1 + e^{-t/\tau} + \cdots + e^{-(n-2)t/\tau}) \tag{1.4.3}$$

where, if they used $\sigma = e^{-t/\tau}$ and, then, set $\alpha = 1/\sigma - 1$, the geometric relationship $G_n = \sigma^{n-1}$ is obtained, as required to support the geometric significance ratio model.

**Remark.** *By a geometric significance function where for all $i$, $\sigma_{i+1}/\sigma_i = \sigma$, $\sigma$ is the geometric significance ratio thus the ordered significance values $\{\sigma_1, \sigma_2, \cdots\}$ form a geometric series.*

## 1.5 Efficient Ranked-NoM (R-NoM) code

We will propose a novel scheme dubbed Ranked-$N$-of-$M$ (R-NoM) coding, that combines features of both ROC and $N$-of-$M$ coding (Figure 1.4) equivalent to Furber implementation 1.4.3 that uses the information contained in the order of only the first inputs but differs in that we add an important property: it is purely integer-based.

Theoretical analysis is an important tool for characterizing what the neural codes do, determining how they function, and understanding why they operate in particular ways. Therefore, we will provide a completely new mathematical framework for comparing the efficiency of spike-based coding schemes where each input only fires a single spike and where information is encoded in the order of firing.

### 1.5.1 Sparse and timing mechanisms implementation

What is the way to encode the information using the timing of the first spike? in principle, you could use the latency at which a single neuron fires in response to an input to derive information about the intensity of the activation. For example, a neurophysiologist could use an oscilloscope to determine a neuron's latency. But this requires knowing precisely when the stimulus came on. Inside the brain, there is no way to know this. Hence, we will consider an alternative strategy: looking across a population of neurons and determining the order in which they fire.

It is worth mentioning that neurons are intrinsically sensitive to the timing of their inputs: shifting the input spike times obviously shifts the response time. But here, we consider additional mechanisms that allow neurons to respond selectively to certain input spike time patterns. For example, [44, 46, 57, 47] used linearly increasing excitatory postsynaptic potentials, such that early spikes contribute more. To obtain a similar effect, [37] used a decaying dendritic kernel. We will focus on spike-based, rather than time-based mechanisms: the input spikes' contribution depends on their arrival ranks rather than on their precise times.

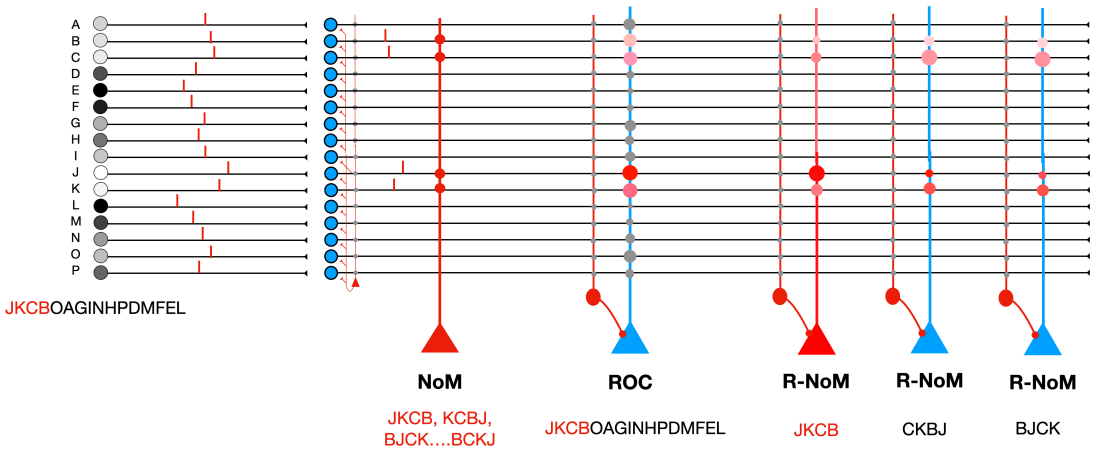



Figure 1.4 – Comparison of different codes. On the left, the $M = 16$ afferents fire in the order JKCBOAGINHPDMFEL, but a $4$-winner-take-all mechanism only lets the $N = 4$ first spikes through. NoM coding: the readout neuron uses binary weights: $W = 4$ ones, and $M - W = 12$ zeros. The final potential reaches the maximal value of $4$ if the $N$ first spikes correspond to the $W$ non-zero weights. The order of these $4$ first spikes does not matter. Rank Order coding **ROC.**: the neuron is set up to respond maximally to the order JKCBOAGINHPDMFEL, even though here only the $4$ input spikes are propagated. Ranked-NoM coding **R-NoM**:: we show three readout neurons that are selective to three different orders for the $4$ first spikes, among the $4! = 24$ possible orders, thanks to graded weights and modulations, both in $\{1, 2, 3, 4\}$.

The idea is always that the first input spikes contribute more, while later input spikes contribute less, or not at all. This is implemented with a modulation function that decreases with the rank, for example, linearly or geometrically. The net contribution of each input spike to the neuron's potential is then the product of the modulation function with the synaptic weight which we called integration $S_{C,I}$. The modulation function can also have a cut-off so that the last spikes make no contribution at all.

## 1.6 Unifying TTFS theory

Our main goal, is to lay the foundation of a mathematical framework in order to assess, from a theoretical point of view, the potential of such order-based TTFS coding schemes.

The unifying mathematical framework will allow the three codes: ROC, $N$-of-$M$ and Ranked-NoM, to be compared in terms of discriminability, which measures to what extent a neuron responds more strongly to its preferred input spike pattern than to random patterns. This unifying theoretical framework also allows tuning the parameters of the codes in order to optimize their discriminative power. As an illustration of this framework, the analysis will be performed upon three instances of such coding schemes: two previous proposals (Rank Order Coding and NoM coding) and our combination of both (Ranked-NoM Coding).

### 1.6.1 Discriminability power measure

We will compare the three coding schemes in terms of discriminative power, characterizing its distribution by the difference between its best possible value and its expected values, scaled by its variance.

**Definition 1.6.1.** We define discriminability $D_C(I)$ as:

$$D_C(I) = \frac{\max(S_{C,I}) - \mathrm{E}[S_{C,I}]}{\sqrt{\mathrm{Var}[S_{C,I}]}} \tag{1.6.1}$$

where $I \in \mathbb{Z}$ and takes values for ROC in the interval $[1, M]$ and for Ranked-NoM and NoM coding in the interval $[1, \mathcal{N}]$. Given that for values $\mathcal{N} < I < M$, Ranked-NoM and NoM are not defined, we set those values to the final integration corresponding to each scheme.

This discriminability is also known as the signal-to-noise ratio in other papers [27, 28, 19], as it is the equivalent of d' (d-prime) measures used in Signal Detection Theory (and psychophysics). A higher index indicates that the signal can be more readily detected [56].

### 1.6.2 Contributions

Our main result is the new mathematical framework which allows rigorous theoretical comparison of a broad range TTFS codes. The overall mathematical formulation for the first two moments of the distribution of potential, at any stage of the propagation, and for the three schemes under consideration has never been done before. Moreover the foundational framework has been designed and explained so as to be used for future studies using other TTFS schemes.

Ranked -NoM (R-NoM) coding scheme is the first proposal which is purely integer-based which leverage sparsity and time features. We demonstrate that R-NoM coding scheme outperforms previous proposals in terms of discriminative power.

These results are published in the Frontiers paper.

### 1.6.3 Constrains

Neuromorphic sensory systems can record dynamic changes in activity from an environment in real-time which can be combined with SNN to enable extremely low-power computing. Researchers in the robotics community have already demonstrated the benefit of using even-based sensors for tracking and gesture recognition, among other applications[36] [42]. However, most of these applications use DNL to perform recognition. For

instance, TTFS is not well suited for dynamic inputs, since coding changes in the input requires additional spikes. We will thus focus on static inputs, e.g. flashed images. For simplicity and hardware-friendliness, we will also restrict ourselves to non-leaky neurons. A leak is useful to process dynamic inputs because the oldest inputs should be forgotten. Yet it is not required with the static inputs used in this work.

A major restriction in the use of SNNs with such sensors is the lack of appropriate training algorithms that can efficiently utilize the timing information of the spiking neurons. Spiking neurons have a discontinuous functionality, and emit discrete spikes that are not differentiable; hence they cannot use the gradient-descent backpropagation techniques that are fundamental to conventional neural network training [43] but see the surrogate gradient approach as a workaround [34].

## 1.7 Chapters planning

We will begin directly with the theoretical development. Chapter $2$ of TTFS coding theory: is an development of the theory from the scratch by systematic and logical flow which will be easy to understand the general theoretical approach. The Chapters $3$, $5$ and $6$ of the application coding schemes are designed to develop the theory of each TTFS coding scheme, ROC, NofM and Ranked-NoM. The three temporal codes has a intuitive meaning. This intuition will be translated into an abstract model to be generalized gradually. Examples will be provided to explain the scores generated function to get the sample space for each scheme to develop the mathematical framework. Thus we will get the parameters of the discriminability measure which let us to compare the codes. The Chapter IV NofM codes Hypergeometric distribution: we will develop a proof of statistics spikes generation when we compare by dot product two NofM codes. We create a technique to find the frequency called "Tree method" which is the based to find the covariance of the TTFS coding schemes.

In the closing chapters: TTFS coding schemes comparation, we will analyse the behavior of discriminability for final potential and the behavior of discriminability during propagation of the schemes. We will explore the speed-accuracy trade-off through simulations. We will also analize the effect of the permutations at the weights by pearson-correlation coefficient and finally we will consider an information theoretic view and characterize the Shannon information content contained in Ranked-NoM code. Thus the last Chapter VIII will be dedicate to conclude this work and propose the perspectives for the futures research works.

# 2

# Time-To-First-Spike: A unifying theory

In this chapter, we establish a unifying theory of the Time-To-First-Spike (TTFS) coding in spiking neural networks.

The first section: we introduce the general structure of the theory from the a random experiment.

The second section: *The Scores Support*, is by far the most important of this chapter. To develop TTFS theory three components are needed; 1) a random experiment, 2) the random variable associated with this experiment and 3) the probability distribution of a discrete variable. The last two components depend on a well defined sample space of the random experiment which we call *Scores support* $\Omega_C$.

The subsections will show the way for the construction of the set $\Omega_C$. In order to build the support based upon probability theory we can start from probability structure of rank order of inputs where we convert ranks into weights in $\Omega$ by an affine transformation. From the weights support $\Omega$ we build the scores support $\Omega_C$ depending of the properties of each scheme $C$.

Having well-defined the scores support $\Omega_C$, in the third section we deduce the formulas to find the expectation vector $\mathrm{E}[\mathbf{W}]$ and the variance-covariance matrix $\mathbf{K}_{WW}$ of the random vector$\mathbf{W} \in \Omega_C$.

In the fourth section, we define mathematically the response of an afferent neurone by defining modulation function and integration function $S_C(\mathbf{w}_C^k, I)$. Given that the integration is a random variable we formulate the first two moments as well as the maximum of the integration and define a measure of discriminability power in the last section.

## 2.1 Random Experiment

We first define a random experiment for the spikes generated by $M$ neurons. For a given stationary stimulus, each of the $M$ input neurones emits one spike. Input patterns will then translate into vectors of size $M$. Each order pattern (vector) is a simple event, for instance

ROC has $M!$ simple events. these events are mutually exclusive. The set of all simple events is the sample space of the experiment.

### 2.1.1 Sample Space

The notion of sample space comes from Ludwig von Mises (he is considered one of the most influential economic and political thinkers of the $20$th century). This notion made it possible to build up a strictly mathematical theory of probability based on measure theory. We will develop the theory based on a well-defined sample space of each scheme which will be denoted as $\Omega_C$.

When $M$ neurons trigger all $M!$ possible order the possible results have the same probability, from a theoretical standpoint, this a convention, we agree to consider them as equally probable. If all points (events) have the same probability, then the probability of an order pattern (event) is the ration of the number of favorable cases to the total numbers of cases.

We denote $\Omega$ the set of the weights vectors, or set of sample spaces called support weights. $\Omega$ is the base to establish the support of each coding scheme. For this, we define, for each scheme $C$, a vector-value function $\Phi_C$ from $\Omega$ to $\Omega_C$ and we use the term *score vector* to denote elements in $\Omega_C$.

By construction, each component of the order pattern $X_i \in \Omega$ is a discrete random variable taking values from the set $\mathcal{D}_{X_i} = \{0, 1, \ldots, M-1\}$ with marginal probability distribution $\mathrm{P}_{X_i}(r) = \frac{1}{M}$, and multivariate joint probability distribution $\mathrm{P}_{X_1 X_2 \ldots X_M} = \frac{1}{M!}$. $X_i$ are identically distributed and they are not independent since realizations of $\mathbf{X}$ are permutations from a unique set of values, the one prescribed by the coding scheme, which implies correlation, so that: Cov$(W_i, W_j) \neq 0$.

Having well defined the random scores vectors for each temporal coding scheme by their scores-support; $\Omega_C$, we establish the probability and statistics to get the first two moments of the scores for each coding scheme.

The theoretical development takes place under multivariate calculus, multivariate probability theory and notions of combinatorial analysis. We will use the fundamental and basic concepts of those brands of mathematics. Thus we will show the mathematical way to get the formulas for the maximum, expected value and variance of each TTFS coding scheme at intermediate states and at the final potential.

## 2.2 The Scores Support $\Omega_C$

### 2.2.1 Preliminaries

We formulate some definitions and theorems in which is based our spike-based coding theory. Let us define the random experiment.

**Definition 2.2.1** (Random Experiment)**.** Let us consider an input layer made of $M$ spiking neurones, denoted $I_0, .., I_{(M-1)}$, which send spikes to a efferent neurone. For a given stationary stimulus, each $M$ input neurone emits one spike, and the order in which spikes are emitted depends on the stimulus.

We encode an order of the input neurones spikes by the vector corresponding to the order of neurones indices. For instance, the order $I_0, .., I_{(M-1)}$ will translate into $0, 1, \ldots, M-1$, and the inverse order will translate into $M-1, M-2, \ldots, 1, 0$.

We will deal with finite samples spaces (possible outcomes) in which we assign equal probabilities to all of events and then speak of random samples. The word "random" is not well defined, but when applied to samples or selections is has a unique meaning: The term random choice is meant to imply that all outcomes are equally probable [8].

Before to define the sample space is worthy, to formulate an important theorem of combinatorial analysis in which we can understand the meaning of sampling and the nature of the combinatorial formulas which is the based to find the probability distributions.

Consider the set or population of $n$ elements $a_1, a_2, \ldots, a_n$. Any ordered arrangement $a_{j_1}, a_{j_2}, \ldots, \ a_{j_r}$ of $r$ symbols is called an ordered sample of size $r$ drawn from our population. Two procedures are then possible. First, *sampling with replacement*; here the same element can be drawn more than once. Second, *sampling without replacement*; here an element once chosen is removed from the population. Thus we have the following main theorem,

**Theorem 2.2.1.** *For a population of $n$ elements and a prescribed sample size $r$, there exist $n^r$ different samples with replacement and $(n)_r$ samples without replacement.*

$$(n)_r = n(n-1)\cdots(n-r+1)$$

We note the special case where $r = n$. In sampling without replacement a sample of size $n$ includes the whole population and represents a reordering(or permutation) of its elements. [8]

**Corollary 2.2.1.** *The number of different orderings of $n$ elements is*

$$n! = n(n-1)\cdots 2 \cdot 1$$

**Remark** (Sample Space-Support)**.** *Each pattern input-spike is a vector of size $M \times 1$. Then the possible outputs of the experiment contains $M!$ vectors, where their components belong to the set $0, 1, ..., M-1$.*

Note that each vector is an event that contains an order in which the neurons trigger. This input order is transformed into a vector of weights. Depending on the scheme, the vector of weights is used to build a vector of scores.

The response of an efferent neurone is then built as a sum of the scores multiplied by a modulation which is specific for each neurone. In others words, the integration random variable is the result of the dot product between scores vectors and a fix modulation vector.

Thus, let us build the theoretical framework for the scores support $\Omega_C$.

### 2.2.2 Input orders

Let $\Lambda$ denote the ascending lexically ordered set of the possible permutations over the set $\mathcal{M} = \{0, \ldots, M-1\}$: $\Lambda = \{\mathbf{r}^1, ..., \mathbf{r}^{M!}\}$ where $\mathbf{r} \in \mathbb{R}^M$ and $r_i^j \in \mathcal{M}$. To map a rank into an input order in $\Lambda$, we define the application $\mathcal{R} : \{1, 2, .., M!\} \to \Lambda$ such that $\mathcal{R}(k) = \mathbf{r}^k$.

We consider the discrete random variable $K$, defined as:

$$K = \begin{cases} \mathcal{D}_K = \{1, 2, .., M!\} \\ \\ \mathrm{P}_K(k) = \frac{1}{M!} \end{cases}$$

Building upon $K$, we can then consider $\mathbf{X} = \mathcal{R}(K) = (X_1, X_2, \ldots, X_M)$ [1] as a random vector or multivariate random variable with support the ordered set $\Lambda$:

$$\Lambda_{X_1 X_2 \ldots X_M} = \left\{ \begin{bmatrix} 0 \\ 1 \\ \vdots \\ M-1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ M-1 \end{bmatrix} \cdots \begin{bmatrix} M-1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \right\} \tag{2.2.1}$$

By construction, it is a vector of size $M$, and each component $X_i$ is a random variable defined on the same sample space obeying:

$$X_i = \begin{cases} \mathcal{D}_{X_i} = \{0, \ldots, M-1\} \\ \\ \mathrm{P}_{X_i}(r_i) = \frac{1}{M} \end{cases}$$

Let its multivariate joint probability mass function be

$$P_{X_1 X_2 \ldots X_M} = \begin{cases} \dfrac{1}{M!} & \mathbf{r}^k \in \Lambda_{X_1 X_2 \ldots X_M} \\ 0 & \text{Otherwise} \end{cases} \tag{2.2.2}$$

$X_1, X_2, \ldots, X_M$ are identically distributed, hence they have same expectation and variance ; they are not independent given that:

$$P_{X_1}(r) P_{X_2}(r) \ldots P_{X_M}(r) = \left(\frac{1}{M}\right)^M \neq \frac{1}{M!} = P_{X_1 X_2 \ldots X_M} \tag{2.2.3}$$

thus Cov$(X_i, X_j) \neq 0$.

### 2.2.3 From input order to weights

**Definition 2.2.2.** We define the affine transformation $\Phi$ mapping vectors from $\Lambda$ (input orders) to the set $\Omega$ (weights):

$$\begin{array}{rccc} \Phi: & \Lambda \subset \mathbb{R}^M & \longrightarrow & \Omega \subset \mathbb{R}^M \\ & (r_1^k, \ldots, r_M^k) & \longrightarrow & (\hat{w}_1^k, \ldots, \hat{w}_M^k) \\ & \mathbf{r}^k & \longrightarrow & \hat{\mathbf{w}}^k \end{array} \tag{2.2.4}$$

[1] Throughout the article, we will use rows or columns interchangeably to denote random variables and define vector-value functions.

Each vectorial component is given by

$$\phi_i(\mathbf{r}^k) = \phi_i(r_1^k, \ldots, r_M^k) = M - r_i^k = \hat{w}_i^k \tag{2.2.5}$$

so we have:

$$\begin{aligned}
\Phi(\mathbf{r}^k) &= \Phi(r_1^k, \ldots, r_M^k) && (2.2.6)\\
&= (\phi_1(r_1^k, \ldots, r_M^k), \phi_2(r_1^k, \ldots, r_M^k), \ldots, \phi_M(r_1^k, \ldots, r_M^k)) && (2.2.7)\\
&= (M - r_1^k, M - r_2^k, \ldots, M - r_M^k) && (2.2.8)\\
&= (\hat{w}_1^k, \ldots, \hat{w}_M^k) = \hat{\mathbf{w}}^k && (2.2.9)
\end{aligned}$$

Now considering $X_1, ..., X_M$ the random variables with joint probability distribution function $P_{X_1 \ldots X_M}(x_1, \ldots, x_M)$ and support $\Lambda$. Let us denote $\hat{W}_i^k = \phi_i(X_1, ..., X_M)$ for $i = 1, \ldots, M$. We transform the random variables $X_i$ (input order) into the new random variables $\hat{W}_i$ (weights) by the affine transformation $\Phi$.

Since the multivariate transformation $\Phi$ is one to one, the transformation is invertible and can be solved for the equations $r_i^k = \phi_i^{-1}(\hat{w}_1^k, \ldots, \hat{w}_M^k)$ for $i = 1, \ldots, M$.

The Jacobian of this multivariate transformation is

$$J\Phi = \begin{vmatrix} \nabla\phi_1 \\ \nabla\phi_2 \\ \vdots \\ \nabla\phi_M \end{vmatrix} = \begin{vmatrix} \frac{\partial\phi_1}{\partial r_1^k} & \frac{\partial\phi_1}{\partial r_2^k} & \cdots & \frac{\partial\phi_1}{\partial r_M^k} \\ \frac{\partial\phi_2}{\partial r_1^k} & \frac{\partial\phi_2}{\partial r_2^k} & \cdots & \frac{\partial\phi_2}{\partial r_M^k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\phi_M}{\partial r_1^k} & \frac{\partial\phi_M}{\partial r_2^k} & \cdots & \frac{\partial\phi_M}{\partial r_M^k} \end{vmatrix} = \begin{vmatrix} -1 & 0 & \ldots & 0 \\ 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -1 \end{vmatrix} = -1 \tag{2.2.10}$$

Let $|J\Phi|$ denote the absolute value of the determinant $J$ . Then the joint pdf of $\hat{W}_1^k, \ldots, \hat{W}_M^k$ is

$$P_{\hat{W}_1 \ldots \hat{W}_M}(\hat{w}_1^k, \ldots, \hat{w}_M^k) = P_{X_1 \ldots X_M}(\phi_1^{-1}(\hat{w}_1^k), \ldots, \phi_M^{-1}(\hat{w}_1^k))|J\Phi| \tag{2.2.11}$$

The inverse for $i = 1, \ldots, M$ is given by

$$\phi_i^{-1} = M - w_i^k = r_i^k \tag{2.2.12}$$

Therefore, $\hat{\mathbf{W}} = (\hat{W}_1, \hat{W}_2 \ldots, \hat{W}_M)$ is a discrete random vector with weights-support the ordered set $\Omega = \{\hat{\mathbf{w}}_1^k, \ldots, \hat{\mathbf{w}}_{M!}^k\}$

$$\Omega_{\hat{W}_1 \hat{W}_2 \ldots \hat{W}_M} = \left\{ \begin{bmatrix} M \\ M-1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} M-1 \\ M \\ \vdots \\ 1 \end{bmatrix} \cdots \begin{bmatrix} 1 \\ \vdots \\ M-1 \\ M \end{bmatrix} \right\} \tag{2.2.13}$$

and its joint pdf is given by

$$\begin{aligned} P_{\hat{W}_1\ldots\hat{W}_M}(\hat{w}_1^k,\ldots,\hat{w}_M^k) &= P_{X_1\ldots X_M}(\phi_1^{-1}(\hat{w}_1^k),\ldots,\phi_M^{-1}(\hat{w}_1^k))|J\Phi| \\ &= P_{X_1\ldots X_M}(\phi_1^{-1}(\hat{w}_1^k),\ldots,\phi_M^{-1}(\hat{w}_1^k)) \\ &= P_{X_1\ldots X_M}(M-w_1^k,\ldots,M-w_M^k) \\ &= \frac{1}{M!} \end{aligned} \tag{2.2.14}$$

In order to find the *marginal probability mass function* of $\hat{W}_1,\hat{W}_2,\ldots,\hat{W}_M$, we will use the fact that the joint probability distribution $P_{\hat{W}_1\ldots\hat{W}_M}(\hat{w}_1^k,\ldots,\hat{w}_M^k)$ contains all the information regarding the marginal distributions of $\hat{W}_1,\hat{W}_2,\ldots,\hat{W}_M$. This means that, we can obtain the probability mass function of $\hat{W}_1$ from its joint probability function with $\hat{W}_2,\ldots,\hat{W}_M$.

**Proposition 2.2.1.** *Let $Y_1,Y_2,\ldots,Y_n$ be discrete random variables defined on the same sample space. The marginal probability function of $\beta$th random variable $Y_\beta$ of $\gamma$th output $y_\gamma$, $P_{Y_\beta}(y_\gamma)$, is given by:*

$$P_{Y_\beta}(y_\gamma) = \sum_{y_{i_n}\in\mathcal{D}_{Y_n}} \cdots \sum_{\substack{y_{i_k}\in\mathcal{D}_{Y_k}\\ k\neq\beta}} \cdots \sum_{y_{i_1}\in\mathcal{D}_{Y_1}} P_{Y_1\ldots Y_\beta\ldots Y_n}(y_{i_1},\ldots,y_\gamma,\ldots,y_{i_n}) \tag{2.2.15}$$

*Proof.* Let us focus on two random variables $X$ and $Y$ with sample spaces $R_X,R_Y$ respectively, and then we generalize to $M$ random variables. We can write

$$P_X(x) = P(X=x) \tag{2.2.16}$$

$$= \sum_{y_j\in R_Y} P(X=x,Y=y_j) \quad \text{law of total probability} \tag{2.2.17}$$

$$= \sum_{y_j\in R_Y} P_{XY}(x,y_j) \tag{2.2.18}$$

Thus the Marginal Probability mass functions of $X$ and $Y$ are given by,

$$P_X(x) = \sum_{y_j\in R_Y} P_{XY}(x,y_j), \quad \forall x\in R_X \tag{2.2.19}$$

$$P_Y(y) = \sum_{x_i\in R_X} P_{XY}(x_i,y), \quad \forall y\in R_Y \tag{2.2.20}$$

Therefore, the extension to $M$ random variables is straightforward. Let $Y_1,Y_2,\ldots,Y_n$ be random variables defined on the same sample space $\mathcal{D}_{Y_j}=\{y_1,y_2,\ldots,y_m\}$ and $P_{Y_1Y_2\ldots Y_n}$ be the multivariate probability distribution of the random variables $Y_j$, then we have:

$$P_{Y_1}(y_1) = \sum_{y_{i_n}\in\mathcal{D}_{Y_n}} \cdots \sum_{y_{i_2}\in\mathcal{D}_{Y_2}} P_{Y_1Y_2\ldots Y_n}(y_1,y_{i_2},\ldots,y_{i_n}) \tag{2.2.21}$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$P_{Y_1}(y_m) = \sum_{y_{i_n}\in\mathcal{D}_{Y_n}} \cdots \sum_{y_{i_2}\in\mathcal{D}_{Y_2}} P_{Y_1Y_2\ldots Y_n}(y_m,y_{i_2},\ldots,y_{i_n}) \tag{2.2.22}$$

and the same way, we can compute the probability of each element of the sample space of $Y_2, \ldots, Y_n$. From here, we have the following notation for the marginal probability distribution of $\beta$th random variable $Y_\beta$ of $\gamma$th output $y_\gamma$, $P_{Y_\beta}(y_\gamma)$,

$$P_{Y_\beta}(y_\gamma) = \sum_{y_{i_n} \in \mathcal{D}_{Y_n}} \cdots \sum_{\substack{y_{i_k} \in \mathcal{D}_{Y_k} \\ k \neq \beta}} \cdots \sum_{y_{i_1} \in \mathcal{D}_{Y_1}} P_{Y_1 \ldots Y_\beta \ldots Y_n}(y_{i_1}, \ldots, y_\gamma, \ldots, y_{i_n}) \tag{2.2.23}$$

□

Therefore, for proposition 2.2.1, the *marginal probability mass function* of $\hat{W}_1, \hat{W}_2, \ldots, \hat{W}_M$ defined on the same sample space $\mathcal{D}_{X_i} = \{M, M-1, ..., 1\}$ for $i = 1, ..., M$, with joint probability distribution defined in the equation 2.2.14, is given by,

$$P_{\hat{W}_i}(w) = \begin{cases} \dfrac{(M-1)!}{M!} = \dfrac{1}{M} & w = 1, 2, \ldots M \\ 0 & \text{Otherwise} \end{cases} \tag{2.2.24}$$

A coding scheme is defined by how weights are transformed to scores, and how these scores are integrated with a modulation function.

### 2.2.4 From weights to scores

To attribute scores to the weights for a given coding scheme, we define the *score vector*, $\mathbf{w}_C^k = (w_1^k, \ldots, w_M^k) \in \mathbb{R}^M$, by the following function $\Phi_C$:

**Definition 2.2.3.** We define the vector-valued function,

$$\begin{array}{rccc} \Phi_C: & \Omega \subset \mathbb{R}^M & \longrightarrow & \Omega_C \subset \mathbb{R}^M \\ & (\hat{w}_1^k, \ldots, \hat{w}_M^k) & \longrightarrow & (w_1^l, \ldots, w_M^l) \\ & \hat{\mathbf{w}}^k & \longrightarrow & \mathbf{w}_C^l \end{array} \tag{2.2.25}$$

such that:

$$\Phi_C(\hat{\mathbf{w}}^k) = \Phi_C(\hat{w}_1^k, \ldots, \hat{w}_M^k) \tag{2.2.26}$$

$$= (\phi_{C,1}(\hat{w}_1^k, \ldots, \hat{w}_M^k), \ldots, \phi_{C,M}(\hat{w}_1^k, \ldots, \hat{w}_M^k)) \tag{2.2.27}$$

$$= (w_1^l, \ldots, w_M^l) = \mathbf{w}_C^l \tag{2.2.28}$$

where $l \in \{1, \ldots, |\Omega_C|\}$ and $\phi_{C,i}$ are real functions of several variables, $\phi_{C,i} : \Omega \to \mathbb{R}$ that are defined depending on the coding scheme.

Therefore, let $\mathbf{W} = (W_1, W_2 \ldots, W_M)$ be a discrete random vector with support the ordered set $\Omega_C = \{\mathbf{w}_1^l, \ldots, \mathbf{w}_{|\Omega_C|}^l\}$ where $\mathbf{w}_C^l$ are the scores of scheme $C$.

Being built upon $\mathbf{W}$, the components $W_i$ of $\mathbf{W}$ are also identically distributed hence they have same expectation and variance. We will show below they are not independent, and will establish their covariance by *tree method*, one for each scheme (**??**)(5.2.8)(3.2.12).

We denote $\Omega_C$ the set of possible values for $\mathbf{w}_C^k$, that is, the scores-support and its cardinality $|\Omega_C|$ depends on the coding scheme.

## 2.3 Scores as a random vectors

For many statistical purposes it is useful to represent higher dimensional random variables in terms of vector-matrix notation. We can write the multivariate random variable $W_1, W_2, \ldots, W_M$ as a column vector, denoted

$$\mathbf{W} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_M \end{bmatrix} \tag{2.3.1}$$

In the next sections we define the expectation vector and the matrix for a random vectors or multivariate random variables.

### 2.3.1 Expectation vector

The expectation of a random vector $\mathbf{W}$ is the vector of expectations of the random variables $W_1, \ldots, W_M$. Thus, the expected value of a random variables $W$

$$\mathrm{E}[W] = \sum_{i=1}^{M} w_i \; P(w_i) \tag{2.3.2}$$

$P(w_i)$ is the marginal probability distribution function. Therefore, the expected value vector or the mean vector of the score random vector $\mathbf{W}$ is defined as

$$\mathrm{E}[\mathbf{W}] = \begin{bmatrix} \mathrm{E}[W_1] \\ \mathrm{E}[W_2] \\ \vdots \\ \mathrm{E}[W_M] \end{bmatrix} \tag{2.3.3}$$

### 2.3.2 Variance-Covariance Matrix

$\mathbf{K}_{WW}$ is the $M \times M$ Variance-Covariance Matrix of scores $\mathbf{W}$:

$$\begin{aligned} \mathbf{K}_{WW} &= \mathrm{E}[(\mathbf{W} - \mathrm{E}[\mathbf{W}])(\mathbf{W} - \mathrm{E}[\mathbf{W}])^{\mathsf{T}}] && (2.3.4) \\ &= \mathrm{E}[\mathbf{W}\ \mathbf{W}^{\mathsf{T}}] - \mathrm{E}[\mathbf{W}]\ \mathrm{E}[\mathbf{W}]^{\mathsf{T}} \\ &= \begin{bmatrix} \mathsf{Cov}(W_1, W_1) & \mathsf{Cov}(W_1, W_2) & \ldots & \mathsf{Cov}(W_1, W_M) \\ \mathsf{Cov}(W_2, W_1) & \mathsf{Cov}(W_2, W_2) & \ldots & \mathsf{Cov}(W_2, W_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{Cov}(W_M, W_1) & \mathsf{Cov}(W_M, W_2) & \ldots & \mathsf{Cov}(W_M, W_M) \end{bmatrix} && (2.3.5) \end{aligned}$$

where covariance $\mathsf{Cov}(W_i, W_j)$ is defined as:

$$\begin{aligned} \mathsf{Cov}(W_i, W_j) &= \mathrm{E}[(W_i - \mathrm{E}[W_i])(W_j - \mathrm{E}[W_j])] && (2.3.6) \\ &= \sum_{i,j} (w_i - \mu_W)(w_j - \mu_W) f(W_i = w_i, W_j = w_j) && (2.3.7) \end{aligned}$$

where $\mu_W = \mathrm{E}[W_i] = \mathrm{E}[W_j]$ and $f(W_i, W_j)$ is the bivariate joint probability distribution.

The diagonal elements are the variance of $W_i$ with $\mathsf{Cov}(W_i, W_i) = \mathrm{E}[(W_i - \mathrm{E}[W_i])^2] = \mathsf{Var}(W_i)$.

#### Properties of $\mathbf{K}_{WW}$ matrix

- Since $\mathbf{W}$ elements are identically distributed, they all have the same variance and the same co-variance

- $\mathrm{Cov}(W_i, W_j) = \mathrm{Cov}(W_j, W_i)$, $\mathbf{K}_{WW}$ is then symmetric and has only two values, an on-diagonal value $\mathrm{Var}(W_i) \equiv \mathrm{Var}W$, and an off-diagonal value $\mathrm{Cov}(W_i, W_j) \equiv \vartheta$, reading:

$$\mathbf{K}_{WW} = \begin{bmatrix} \mathrm{Var}W & \vartheta & \dots & \vartheta \\ \vartheta & \mathrm{Var}W & \dots & \vartheta \\ \vdots & \vdots & \ddots & \vdots \\ \vartheta & \vartheta & \dots & \mathrm{Var}W \end{bmatrix} \tag{2.3.8}$$

  We denote the variance-covariance matrix for each $C$ scheme as $\mathbf{K}^C_{WW}$

- $\mathbf{K}^C_{WW}$ is positive-semidefinite. Let us define the random variable

$$X = \mathbf{a}^T(\mathbf{W} - \mathrm{E}[\mathbf{W}]) \tag{2.3.9}$$

  where $\mathbf{a}$ is any $M \times 1$ fix vector. We have that,

$$\mathrm{E}[X^2] \geq 0 \tag{2.3.10}$$
$$\mathrm{E}[XX^T] \geq 0 \tag{2.3.11}$$
$$\mathrm{E}[(\mathbf{a}^T(\mathbf{W} - \mathrm{E}[\mathbf{W}]))((\mathbf{W} - \mathrm{E}[\mathbf{W}])^T\mathbf{a})] \geq 0 \tag{2.3.12}$$
$$\mathbf{a}^T\,\mathrm{E}[(\mathbf{W} - \mathrm{E}[\mathbf{W}])(\mathbf{W} - \mathrm{E}[\mathbf{W}])^T]\mathbf{a} \geq 0 \tag{2.3.13}$$

  therefore,

$$\mathbf{a}^T\,\mathrm{K}_{\mathbf{WW}}\,\mathbf{a} \geq 0 \quad \text{for all} \quad \mathbf{a} \in \mathbb{R}^M \tag{2.3.14}$$

## 2.4 Response of an efferent neurone-Integration

### 2.4.1 Modulations vector function

In order to sort out among stimuli by output neurones, vectors of modulation values must be defined with same size as $\mathbf{w}^k_C$. In the same way as we defined $\mathbf{w}^k$, modulation vectors $\mathbf{v}^l$ can be defined by a modulation function following:

**Definition 2.4.1.** We define the application,

$$\begin{array}{rccc} \Psi_C: & \Omega \subset \mathbb{R}^M & \longrightarrow & \Xi_C \subset \mathbb{R}^M \\ & (\hat{w}^k_1, \dots, \hat{w}^k_M) & \longrightarrow & (v^l_1, \dots, v^l_M) \\ & \hat{\mathbf{w}}^k & \longrightarrow & \mathbf{v}^l_C \end{array} \tag{2.4.1}$$

such that:

$$\Psi_C(\hat{\mathbf{w}}^k) = \Psi_C(\hat{w}^k_1, \dots, \hat{w}^k_M) \tag{2.4.2}$$
$$= (\psi_{C,1}(\hat{w}^k_1, \dots, \hat{w}^k_M), \dots, \psi_{C,M}(\hat{w}^k_1, \dots, \hat{w}^k_M)) \tag{2.4.3}$$
$$= (v^l_1, \dots, v^l_M) = \mathbf{v}^l_C \tag{2.4.4}$$

where $l \in \{1, \ldots, |\Xi|\}$ and $\psi_{C,i}$ are real functions of several variables, $\psi_{C,i} : \Omega \to \mathbb{R}$ that depend upon the coding scheme.

We denote $\Xi_C$ the set of possible values for $\mathbf{v}_C^k$ and its cardinality depends on the coding scheme through $\Psi_C$.

Modulation vector will determine which input order is preferred by the efferent neurone under consideration. For the sake of clarity, we will consider from now on only the output neurone for which the preferred stimulus (input order) is the one corresponding to $\mathbf{r}^1$. Hence we will use

$$\mathbf{v}_C^1 = \Psi_C(\hat{\mathbf{w}}^1) = \Psi_C(\Phi(\mathbf{r}^1)) \tag{2.4.5}$$

as the modulation vector.

### 2.4.2 Integration function $S_C(\mathbf{w}_C^k, I)$

To compare how *scores vector* matches well with *modulations vector*, we define an *integration function* following:

**Definition 2.4.2.** We define the application $S_C(\mathbf{w}_C^k, I) : \Omega_C \times \{0, \ldots, M\} \to \mathbb{R}$ such as:

$S_C(\mathbf{w}_C^k, I)$ is the inner product of *modulations vector* $\mathbf{v}_C^1$ (set for best matching $\hat{\mathbf{w}}^1$ for the coding scheme $C$) by *score vector* $\mathbf{w}_C^k$, over the $I$ first components of vectors.

In order to formally translate intermediate states, we first define the gate function $G_I : \Xi_C \to \mathbb{R}^M$ which nullifies all component of modulations vector for ranks beyond $I$.

Then, integration function reads:

$$S_C(\mathbf{w}_C^k, I) = \left\langle G_I\left(\Psi_C(\hat{\mathbf{w}}^1)\right), \mathbf{w}_C^k \right\rangle \tag{2.4.6}$$

where $G_I\left(\Psi_C(\hat{\mathbf{w}}^1)\right) = G_I\left(\mathbf{v}_C^1\right) = \mathbf{v}_{C,I}^1$. We used bracket notation for inner product.

Building upon the random variable $\mathbf{W}$, we finally define the random variable $S_{C,I} = S_C(\mathbf{W}, I)$:

$$S_{C,I} = \begin{cases} \mathcal{D}_{S_{C,I}} = \{depends\ upon\ C\} \\ \\ \mathrm{P}_{S_{C,I}} = \{depends\ upon\ C\} \end{cases}$$

We can then define the best order $\max(S_{C,I})$, expectation $\mathrm{E}[S_{C,I}]$ and variance $\mathrm{Var}[S_{C,I}]$ for each coding scheme.

### 2.4.3 The first two moments of the response, $\mathrm{E}[S_{C,I}]$

The expectation $\mathrm{E}[S_{C,I}]$ and variance $\mathrm{Var}[S_{C,I}]$ of integration of each scheme $C$ depend on the mean $\mu_{W_C}$, variance $\mathrm{Var}W_C$ and covariance $\mathrm{Cov}_C(W_i, W_j)$ of the scores for the corresponding coding scheme $C$.

As a general pattern, we have the following linear combinations,

$$\mathrm{E}[S_{C,I}] = \lambda_C\ \mu_{W_C} \tag{2.4.7}$$

$$\mathrm{Var}[S_{C,I}] = \alpha_C\ \mathrm{Var}W_C + \beta_C\ \mathrm{Cov}_C(W_i, W_j) \tag{2.4.8}$$

where the constants $\lambda_C, \alpha_C$ and $\beta_C$ for each scheme depend on the modulation vector parameters.

#### Expectation of $S_{C,I}$

For the sake of clarity, let $\mathbf{v}_{C,I} = \mathbf{v}_{C,I}^1 = G_I\left(\Psi_C(\mathbf{r}^1)\right)$ denote the modulation vector, gated up to the first I components.

We have:

$$\begin{aligned} \mathrm{E}[S_{C,I}] &= \mathrm{E}[S_C(\mathbf{W}, I)] \\ &= \mathrm{E}\left[\langle \mathbf{v}_{C,I}, \mathbf{W} \rangle\right] \\ &= \mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathrm{E}\left[\mathbf{W}\right] \end{aligned} \tag{2.4.9}$$

where the expected value of a random vector is the vector whose elements are the expected values of the respective random variables.

#### Variance of $S_{C,I}$

Using same notation $\mathbf{v}_{C,I}$, variance of $S_C(K, I)$ reads:

$$\mathrm{Var}\left[S_{C,I}\right] = \mathrm{Var}\left[S_C(\mathbf{W}, I)\right] \tag{2.4.10}$$

$$= \mathrm{Var}\left[\mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathbf{W}\right] \tag{2.4.11}$$

$$= \mathrm{E}[\mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathbf{W} \cdot \mathbf{W}^{\mathsf{T}} \cdot \mathbf{v}_{C,I}] - \mathrm{E}[\mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathbf{W}] \cdot \mathrm{E}[\mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathbf{W}]^{\mathsf{T}} \tag{2.4.12}$$

$$= \mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathrm{E}[\mathbf{W} \cdot \mathbf{X}^{\mathsf{T}}] \cdot \mathbf{v}_{C,I} - \mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathrm{E}[\mathbf{W}] \cdot \mathrm{E}[\mathbf{W}]^{\mathsf{T}} \cdot \mathbf{v}_{C,I} \tag{2.4.13}$$

$$= \mathbf{v}_{C,I}^{\mathsf{T}} \cdot (\mathrm{E}[\mathbf{W} \cdot \mathbf{W}^{\mathsf{T}}] - \mathrm{E}[\mathbf{W}] \cdot \mathrm{E}[\mathbf{W}]^{\mathsf{T}}) \cdot \mathbf{v}_{C,I} \tag{2.4.14}$$

$$= \mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{C,I} \tag{2.4.15}$$

where $\mathbf{K}_{WW}$ is the $M \times M$ Variance-Covariance Matrix of $\mathbf{W}$ defined in 2.3.8. Note that the variance of integration $\mathrm{Var}\left[S_{C,I}\right]$ must be nonnegative for all choices of the modulation vector $\mathbf{v}_{C,I}$. Since the variance can be expressed as

$$\mathrm{Var}\left[S_{C,I}\right] = \mathbf{v}_{C,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{C,I} \tag{2.4.16}$$

we have that the covariance matrix $\mathbf{K}_{WW}$ must be positive semidefinite. In fact, it is a very important property of the variance-covariance matrix $\mathbf{K}_{WW}$ which has been proved in 2.3.14.

### 2.4.4 Maximum response

#### max($S_{C,I}$)

The rearrangement inequality states that

$$\begin{pmatrix} x_n \\ \vdots \\ x_1 \end{pmatrix}^{\mathsf{T}} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \leq \begin{pmatrix} x_{\sigma(1)} \\ \vdots \\ x_{\sigma(n)} \end{pmatrix}^{\mathsf{T}} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \leq \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}^{\mathsf{T}} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{2.4.17}$$

for every choice of real numbers $x_1 \leq \cdots \leq x_n$ and $y_1 \leq \cdots \leq y_n$ and every permutation $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ of $x_1, \ldots, x_n$.

Then the lower bound is attained only for the permutation which reverses the order, that is, $\sigma(i) = n - i + 1$ for all $i = 1, \ldots, n$, and the upper bound is attained only for the identity, that is, $\sigma(i) = i$.

From equation (2.4.6), the rearrangement inequality yields that $\max(S_{C,I})$ is given by:

$$\max(S_{C,I}) =< \mathbf{v}_{C,I}^1, \mathbf{w}_C^1 > \tag{2.4.18}$$

# 3
# Rank Order Coding

Rank Order Coding (ROC) was an early proposal([49], Fig.3.1). In this work a mechanism for decoding the order of synapse activation that to make the units sensitive to the order of activation of their inputs was proposed. Neurons are maximally activated when inputs arrive in the order of their synaptic weights, thanks to a shunting inhibition mechanism that progressively desensitizes the neurons as spikes arrive.

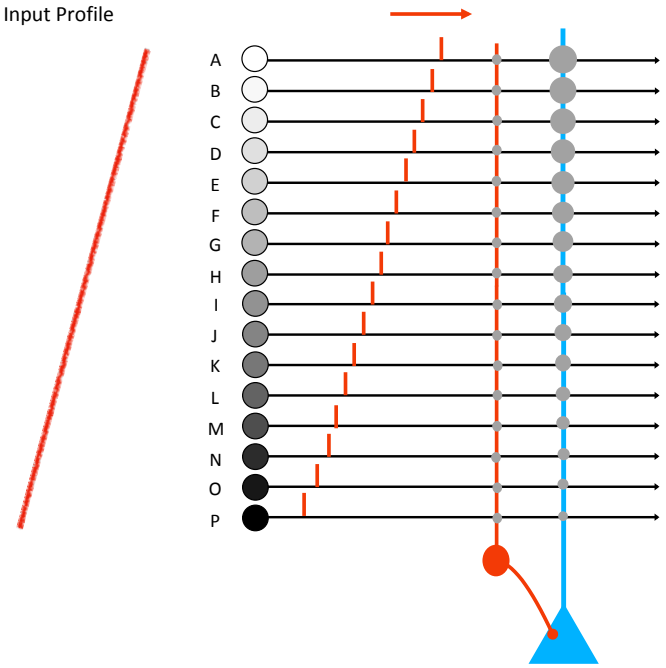



Figure 3.1 – ROC with $M = 16$ afferents. All the afferents fire exactly one spike. Here we show a neuron selective to the input spike order: A,B,$\cdots$, P. Its synaptic weights are linearly decreasing: $M$ for input A, $M-1$ for input B, and so on, down to $1$ for input P. The modulation could be implemented with shunting inhibition, as shown with the red inhibitory neuron.

Synapses will therefore be governed by two values. The synaptic weight, and the efficiency of this weight, determined by its activation rank, which depends on the stimulus. They specified the mechanism of the modulation of synaptic weights by the activation rank of synapses. The modulation function decreases geometrically with the input spike rank.

The statistical approximation of the behaviour of the sum of the weights modulated by their rank $S(n)$, (which we call integration) is based on linear rank statistics theory. (see chapter III). The two first moments of $S(n)$ are computable by rank statistics because they only depend on the marginal moments of the synaptic weights and the modulation.

In this chapter we apply the TTFS coding theory (chapter IV) for ROC to establish the expectation and the variance for the integration at final potential and intermediate states. There are two stages, the first to develop the probabilistic theory of synaptic weights defined as random vectors $\mathbf{W}$ which we call **ROC scores**. In section 3.1 we define the support of the scores $\Omega_R$ to find the joint $P_{W_1 W_2 \dots W_M}$ and the marginal $P_{W_i}(w)$ probability mass functions. We can start directly at the scores thanks to the construction done in Section $1$ of Chapter IV where the rank order is already implicit.

In the section 3.2 we establish the expectation vector $\mathrm{E}[\mathbf{W}]$ and the variance-covariance matrix $\mathbf{K}_{WW}^{H}$. Since the random variables $W_1, W_2, \dots, W_M$ are not independent we find the covariance $\vartheta$ using a tree method to calculate the frequency to bivariate probability distribution $f(W_i, W_j)$.

The second stage is to develop the statistical theory for the random variable integration $S_{R,I}$ at final $I = M$ and intermediate states $I < M$ which consists in computing scores vectors with modulation vector $\mathbf{v}_R^l$ by the inner product. Finally we define a formula for the maximum integration value, $\max(S_{R,M})$, which will use to compare the codes by discriminability power.

## 3.1 ROC scores-Support $\Omega_R$

### 3.1.1 Scores vector function, $\Phi_R$

Rank-Order Coding weights have no other parameter than $M$.

Under this coding, we want the best weights vector $\mathbf{w^1} = \hat{\mathbf{w}}^{\mathbf{1}} = \Phi(\mathcal{R}(1))$ made up of an arithmetic sequence from $M$ down to zero, with rate $-1$.

The weights are obtained by the vector-value function $\Phi_R(\hat{\mathbf{w}}^k) = \mathbf{w}_R^k$ define in (2.2.3). We then define $\phi_{R,i}$ as the identity function,

$$\phi_{R,i}(\hat{\mathbf{w}}^k) = \hat{\mathbf{w}}^k = \mathbf{w}_R^k \tag{3.1.1}$$

such that:

$$\begin{aligned} \Phi_R(\hat{\mathbf{w}}^k) &= \Phi_R(M - r_1^k, M - r_2^k, \dots, M - r_M^k) && (3.1.2)\\ &= (M - r_1^k, M - r_2^k, \dots, M - r_M^k) && (3.1.3)\\ &= (w_1^k, \dots, w_M^k) = \mathbf{w}_R^k && (3.1.4) \end{aligned}$$

Note that, for $k = 1$, we have that $\hat{\mathbf{w}}^1 = (M, M-1 \dots, 1)$, given that $\mathbf{r}^1 = (r_1^1, \dots, r_M^1) =$

$(0, 1, \ldots, M-1)$. Thus

$$\Phi_R(\hat{\mathbf{w}}^1) = \Phi_R(M, M-1, \ldots, 1) \tag{3.1.5}$$
$$= (M, M-1, \ldots, 1) = (w_1^1, \ldots, w_M^1) = \mathbf{w}_R^1 \tag{3.1.6}$$

For $k = M!$, given that $\mathbf{r}^{M!} = (r_1^{M!}, r_2^{M!}, \ldots, r_M^{M!}) = ((M-1), (M-2), \ldots, 0)$ we have that $\hat{\mathbf{w}}^{M!} = (1, 2, \ldots, M)$, so we get:

$$\Phi_R(\hat{\mathbf{w}}^{M!}) = \Phi_R(1, 2, \ldots, M) \tag{3.1.7}$$
$$= (1, 2, \ldots, M) = (w_1^{M!}, \ldots, w_M^{M!}) = \mathbf{w}_R^{M!} \tag{3.1.8}$$

$\Phi_R$ being a bijection, we have:

$$|\Omega| = |\Omega_R| = M! \tag{3.1.9}$$

**Example 3.1.1.** As an illustration, let $M = 4$ ($|\Omega_C| = 24$ permutations). We would have, for the best order:

$$\Phi_R(\hat{\mathbf{w}}^1) = \Phi_R(4, 3, 2, 1) = (4, 3, 2, 1) = \mathbf{w}_R^1 \tag{3.1.10}$$

and for the worst order

$$\Phi_R(\hat{\mathbf{w}}^{24}) = \Phi_R(1, 2, 3, 4) = (1, 2, 3, 4) = \mathbf{w}_R^{24} \tag{3.1.11}$$

### 3.1.2 Probability distributions functions

Let $\mathbf{W} = (W_1\ W_2\ \ldots\ W_M)^{\mathsf{T}}$ be the discrete random vector with support the ordered set $\Omega_R = \{\mathbf{w}^1, \ldots, \mathbf{w}^{M!}\}$, which is generated by the vector-value function $\hat{\Phi}_R$ defined in (3.1.1), therefore,

$$\Omega_R = \Omega_{W_1 W_2 \ldots W_M} = \left\{ \begin{bmatrix} M \\ M-1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} M-1 \\ M \\ \vdots \\ 1 \end{bmatrix} \cdots \begin{bmatrix} 1 \\ \vdots \\ M-1 \\ M \end{bmatrix} \right\} \tag{3.1.12}$$

Due to (3.1.9), its *joint probability mass function* is

$$P_{W_1 W_2 \ldots W_M} = \begin{cases} \dfrac{1}{M!} & \mathbf{w}^k \in \Omega_{W_1 W_2 \ldots W_M} \\ 0 & \text{Otherwise} \end{cases} \tag{3.1.13}$$

and from (2.2.1), the *marginal probability mass function* of $W_1, W_2, \ldots, W_M$ with sample space $\mathcal{D}_{W_i} = \{M, M-1, ..., 1\}$ for $i = 1, ..., M$, is given by:

$$P_{W_i}(w) = \begin{cases} \dfrac{(M-1)!}{M!} = \dfrac{1}{M} & w = 1, \ldots, M \\ 0 & \text{Otherwise} \end{cases} \tag{3.1.14}$$

## 3.2 Scores Random Vectors, $\mathbf{W}$

### 3.2.1 $\mathrm{E}[\mathbf{W}]$. Expectation vector

**$\mu_W$, expected value of $W_i$**

$$\mathrm{E}[W] = \sum_{i=1}^{M} w_i \; P(w_i) = \sum_{i=1}^{M} w_i \; \frac{1}{M} \tag{3.2.1}$$

$$= \frac{1}{M} \sum_{i=1}^{M} w_i = \frac{1}{M} \frac{M(M+1)}{2} = \frac{(M+1)}{2} \tag{3.2.2}$$

**$\mathrm{E}[\mathbf{W}]$, expectation of $\mathbf{W}$**

Therefore, the expectation of a random vector $\mathbf{W}$ is:

$$\mathrm{E}[\mathbf{W}] = \begin{bmatrix} E[W_1] \\ E[W_2] \\ \vdots \\ E[W_M] \end{bmatrix} = \begin{bmatrix} \frac{M+1}{2} \\ \vdots \\ \frac{M+1}{2} \end{bmatrix} \tag{3.2.3}$$

### 3.2.2 $\mathbf{K}_{WW}^{R}$. Variance-Covariance Matrix

The matrix $\mathbf{K}_{WW}^{H}$ is defined in (2.3.8) then we will find the variance Var$W$ and the covariance Cov$(W_i, W_j) = \vartheta$.

**Var$W$ , variance of $W_i$**

$$\mathrm{Var}(W) = \mathrm{E}[W^2] - \mathrm{E}[W]^2 \tag{3.2.4}$$

$$= \sum w_i^2 P(w_i) - \frac{(M+1)^2}{4} \tag{3.2.5}$$

$$= \sum w_i^2 \frac{1}{M} - \frac{(M+1)^2}{4}$$

$$= \frac{1}{M} \left( \frac{M(M+1)(2M+1)}{6} \right) - \frac{(M+1)^2}{4} \tag{3.2.6}$$

$$= \frac{M^2 - 1}{12} \tag{3.2.7}$$

thus the variance of random variable $W$ is given by,

$$\mathrm{Var}W = \frac{M^2 - 1}{12} = \frac{\mu_W \; (M-1)}{6} \tag{3.2.8}$$

### $\vartheta$, covariance of $W_i, W_j$

From (3.1.13) and (3.1.14), the score random variable $W_i$ of ROC coding are not independent because:

$$\prod_{i=1}^{M} P_{W_i}(w) = \left(\frac{1}{M}\right)^M \neq \frac{1}{M!} = P_{W_1 W_2 \dots W_M} \tag{3.2.9}$$

hence Cov$(W_i, W_j) \neq 0$.

From (2.3.7),we must find the bivariate joint probability distribution $f(W_i = w_i, W_j = w_j), i, j = 1, 2, \dots, M$ to find the covariance. We will use the following facts:

- Each random variable can take on $M$ same values: $1, 2, \dots, M$
- For any combination of $w_i, w_j$ with $i, j = 1, \dots M$ the joint probability distribution $f$ are the same.
- The support $\Omega_R$ which is the sample space consists of $|\Omega_R| = M!$ possible pair outcomes that are equally likely, that is, the probability for each outcome is $\frac{1}{|\Omega_R|}$.
- $f(W_i, W_j) = f(W_j, W_i)$
- $f(W_i = w_i, W_j = w_j) = 0$, for $i = j$

To obtain the support $\Omega_R$ (3.1.12) as a matrix where the rows would represent the random variables $W_1, W_2, \dots, W_M$, we consider all permutations of the set $(M, M-1, \dots, 1)$, and arrange them as columns in decreasing lexical order.

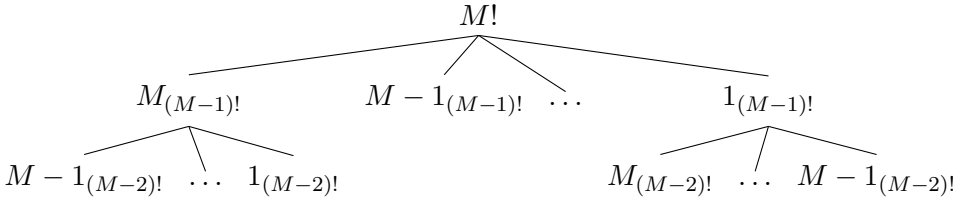
For example, for $M = 3$ the support represented as a matrix is given by:

$$\Omega_1 = \begin{bmatrix} 3 & 3 & 2 & 2 & 1 & 1 \\ 2 & 1 & 3 & 1 & 3 & 2 \\ 1 & 2 & 1 & 3 & 2 & 3 \end{bmatrix}_{3 \times 3!} \tag{3.2.10}$$

and for $M = 4$ we have that,

$$\Omega_2 = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & \dots \\ 3 & 3 & 2 & 2 & 1 & 1 & \dots \\ 2 & 1 & 3 & 1 & 3 & 2 & \dots \\ 1 & 2 & 1 & 3 & 2 & 3 & \dots \end{bmatrix}_{4 \times 4!} \tag{3.2.11}$$

Let us consider the two first variables $W_1, W_2$. If we fix the value for $W_1$, we are left with the permutations of the $M-1$ remaining values. Hence, each value appears $(M-1)!$ times. If we now fix also the value for $W_2$, we are left with the permutations of the $M-2$ remaining values. Hence, we have :




*Tree for ROC coding* (3.2.12)

Therefore, the frequency is the same for each possible pair outcome, and $f = (M-2)!$. Then the probability $P$ is given by:

$$P = \frac{(M-2)!}{M!} = \frac{(M-2)!}{(M-2)!\,(M-1)\,M} = \frac{1}{(M-1)\,M} \tag{3.2.13}$$

We summarize the distribution $f$ in the following table,

| $f(W_i, W_j)$ | 1 | 2 | ... | $M$ | $f_{W_j}(W_j)$ |
|---|---|---|---|---|---|
| 1 | 0 | $P$ | ... | $P$ | $(M-1)P$ |
| 2 | $P$ | 0 | ... | $P$ | $(M-1)P$ |
| $\vdots$ | $\vdots$ | $\vdots$ | ... | $\vdots$ | $\vdots$ |
| $M$ | $P$ | $P$ | ... | 0 | $(M-1)P$ |
| $f_{W_i}(w_i)$ | $(M-1)P$ | $(M-1)P$ | ... | | 1 |

(3.2.14)

To calculate the covariance defined in the equation (2.3.7), we do the operations by rows $c_1, \ldots, c_M$ from the bivariate joint distribution, and obtain:

$$\begin{aligned}
c_1 &= (1-\mu_W)(2-\mu_W)\;P + (1-\mu_W)(3-\mu_W)\;P + \ldots + (1-\mu_W)(M-\mu_W)\;P \\
c_2 &= (2-\mu_W)(1-\mu_W)\;P + (2-\mu_W)(3-\mu_W)\;P + \ldots + (2-\mu_W)(M-\mu_W)\;P \\
&\ldots \\
c_M &= (M-\mu_W)(1-\mu_W)\;P + \ldots + (M-\mu_W)((M-1)-\mu_W)\;P
\end{aligned}$$

Therefore $\text{Cov}(W_i, W_j) = c_1 + c_2 + \ldots + c_M$,

$$\begin{aligned}
\text{Cov}(W_i, W_j) =& (1-\mu_W)\;P[(2-\mu_W) + \ldots + (M-\mu_W)] + \\
& (2-\mu_W)\;P[(1-\mu_W) + (3-\mu_W)\ldots + (M-\mu_W) + \\
& (M-\mu_W)\;P[(1-\mu_W) + (2-\mu_W)\ldots + ((M-1)-\mu_W)]
\end{aligned} \tag{3.2.15}$$

$$\begin{aligned}
=& (1-\mu_W)\;P\left[\frac{M(M+1)}{2} - 1 - (M-1)\mu_W\right] + \\
& (2-\mu_W)\;P\left[\frac{M(M+1)}{2} - 2 - (M-1)\mu_W\right] + \\
& (M-\mu_W)\;P\left[\frac{M(M+1)}{2} - M - (M-1)\mu_W\right]
\end{aligned} \tag{3.2.16}$$

$$= \sum_{i=1}^{M} (i-\mu_W)\;P\,[M\;\mu_W - i - (M-1)\mu_W] \tag{3.2.17}$$

$$= \sum_{i=1}^{M} (i-\mu_W)\;P\;(\mu_W - i) = -P\;\sum_{i=1}^{M} (i-\mu_W)^2 \tag{3.2.18}$$

Using the summation formulas and simplifying, we finally get that the covariance of the scores for the ROC coding is given by:

$$\text{Cov}(W_i, W_j) = \frac{\mu_W}{M-1}\left(\mu_W - \frac{2M+1}{3}\right) \tag{3.2.19}$$

## 3.3 Response of efferent neurone-Integration, $S$

### 3.3.1 Modulation vector function, $\Psi_R$

Under geometrical-ROC that we consider in this paper, we want the modulations vector for $\mathbf{r}^1$ being composed as a decreasing geometric sequence, starting from 1, and with rate $m$.

For (2.4.1), we denote $\Psi_R : \Omega \to \Xi_R$, the function for generating modulations vector for ROC.

It is parametized by the numbers $M \in \{1, \ldots, M\}$ and $m \in [0, 1]$.

We then define $\psi_{R,i}$ as:

$$\psi_{R,i}(\hat{\mathbf{w}}^k; m) = \psi_{R,i}((\hat{w}_1^k, \ldots, \hat{w}_M^k); m) = m^{M-\hat{w}_i^k} \tag{3.3.1}$$

such that

$$\begin{aligned}
\Psi_R(\hat{\mathbf{w}}^k) &= \Psi_R(\hat{w}_1^k, \ldots, \hat{w}_M^k) && (3.3.2)\\
&= (\psi_{R,1}(\hat{w}_1^k, \ldots, \hat{w}_M^k), \ldots, \psi_{R,M}(\hat{w}_1^k, \ldots, \hat{w}_M^k)) && (3.3.3)\\
&= (m^{M-\hat{w}_1^k}, \ldots, m^{M-\hat{w}_M^k}) = \mathbf{v}_R^l && (3.3.4)
\end{aligned}$$

with $l \in \{1, \ldots |\Omega_R|\}$.

$\Psi_R$ is a bijective function so $|\Omega_R| = |\Lambda| = M!$.

For $k = 1$, we have that $\mathbf{r}^1 = (r_1^1\ r_2^1\ \ldots\ r_M^1)^T = (1\ 2\ \ldots\ M)^T$ then

$$\begin{aligned}
\Psi_R(\hat{\mathbf{w}}^1) &= (\psi_{R,1}(\hat{w}_1^1, \ldots, \hat{w}_M^1), \ldots, \psi_{R,M}(\hat{w}_1^1, \ldots, \hat{w}_M^1)) && (3.3.5)\\
&= (m^{M-\hat{w}_1^1}, \ldots, m^{M-\hat{w}_M^1}) && (3.3.6)\\
&= (m^{M-M}, \ldots, m^{M-1}) && (3.3.7)\\
&= (m^0, m^1 \ldots, m^{M-1}) = \mathbf{v}_R^1 && (3.3.8)
\end{aligned}$$

Below, we consider the modulation vector being : $\mathbf{v}_{R,M}^1 = \Psi_R(\hat{\mathbf{w}}^1; m)$.

### 3.3.2 $S_{R,M}$. Integration-Final Potential $I = M$

$\mathrm{E}[S_{R,M}]$. **Expected value**

$$\begin{aligned}
\mathrm{E}[S_{R,M}] &= \mathrm{E}[\mathbf{v}_{R,M}^{\mathsf{T}}\ \mathbf{W}] = \mathbf{v}_{R,M}^{\mathsf{T}}\,\mathrm{E}[\mathbf{W}] && (3.3.9)\\
&= \mathbf{v}_{R,M}^{\mathsf{T}} \cdot \left(\frac{M+1}{2}, ..., \frac{M+1}{2}\right)^{\mathsf{T}} && (3.3.10)\\
&= (m^0\ m^1\ \ldots\ m^{M-1}) \cdot \left(\frac{M+1}{2}, ..., \frac{M+1}{2}\right)^{\mathsf{T}} && (3.3.11)\\
&= \left(\frac{m^0(M+1)}{2} + \frac{m^1(M+1)}{2} + ... + \frac{m^{M-1}(M+1)}{2}\right)\\
&= \frac{(M+1)}{2}(m^0 + m^1 + ... + m^{M-1}) && (3.3.12)\\
&= \frac{(M+1)}{2}\frac{1-m^M}{1-m} && (3.3.13)
\end{aligned}$$

Then the expectation of Integration $S_{R,M}$ is given by

$$\mathrm{E}[S_{R,M}] = \frac{(M+1)}{2}\frac{1-m^M}{1-m} \tag{3.3.14}$$

$\mathrm{Var}[S_{R,M}]$. **Variance**

From (2.4.15), we have $\mathrm{Var}[S_{R,M}] = \mathbf{v}_{R,M}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{R,M}$. The Variance-covariance matrix, $\mathbf{K}_{WW}$ is given by (2.3.8), and the modulation vector $\mathbf{v}_{R,M}^{1}$ of final potential for geometric ROC coding is given by (3.3.8). Therefore:

$$\begin{aligned}
&\mathrm{Var}[S_{R,M}] \\
&= \mathbf{v}_{R,M}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{R,M} \\
&= (m^0\ m^1\ \ldots\ m^{M-1}) \begin{bmatrix} \mathsf{Var}W & \vartheta & \ldots & \vartheta \\ \vartheta & \mathsf{Var}W & \ldots & \vartheta \\ \vdots & \vdots & \ddots & \vdots \\ \vartheta & \vartheta & \ldots & \mathsf{Var}W \end{bmatrix} \begin{bmatrix} m^0 \\ m^1 \\ \vdots \\ m^{M-1} \end{bmatrix} && (3.3.15) \\
&= m^0(m^0\mathsf{Var}W + m^1\vartheta + \ldots + m^M\vartheta) + m^1(m^0\vartheta + m^1\mathsf{Var}W + \ldots + m^M\vartheta) + && (3.3.16) \\
&\ldots + m^{M-1}(m^0\vartheta + m^1\vartheta + \ldots + m^{M-1}\mathsf{Var}W) \\
&= \mathsf{Var}W \sum_{i=0}^{M-1} m^{2i} + && (3.3.17) \\
&\vartheta\left[m^0\left(\sum_{i=0}^{M-1} m^i - m^0\right) + m^1\left(\sum_{i=0}^{M-1} m^i - m^1\right) + m^{M-1}\left(\sum_{i=0}^{M-1} m^i - m^{M-1}\right)\right] \\
&= \left(\frac{1-m^{2M}}{1-m^2}\right)\ \mathsf{Var}W + \left[\sum_{i=0}^{M-1} m^i\left(\frac{1-m^M}{1-m} - m^i\right)\right]\vartheta && (3.3.18)
\end{aligned}$$

Using the geometric summation formulas, we get:

$$\mathrm{Var}[S_{R,M}] = \left(\frac{1-m^{2M}}{1-m^2}\right)\ \mathsf{Var}W + \left[\left(\frac{1-m^M}{1-m}\right)^2 - \frac{1-m^{2M}}{1-m^2} -\right]\ \mathsf{Cov}(W_i, W_j) \tag{3.3.19}$$

where Var$W$ is given by (3.2.8) and Cov$(W_i, W_j)$ by (3.2.19).

### 3.3.3 $S_{R,I}$. Integration-Intermediate states, $I < M$

$\mathrm{E}[S_{R,I}]$, **expectation at intermediate states**

$$\begin{aligned}
\mathrm{E}[S_{R,I}] &= \mathrm{E}[\mathbf{v}_{R,I}^{\mathsf{T}}\ \mathbf{W}] = \mathbf{v}_{R,I}^{\mathsf{T}}\,\mathrm{E}[\mathbf{W}] \qquad (3.3.20)\\
&= \mathbf{v}_{R,I}^{\mathsf{T}} \cdot \left(\frac{(M+1)}{2}, ..., \frac{(M+1)}{2}\right)^{\mathsf{T}}\\
&= (m^0\ m^1\ \ldots\ m^{I-1}\ 0 \ldots 0) \cdot \left(\frac{(M+1)}{2}, ..., \frac{(M+1)}{2}\right)^{\mathsf{T}}\\
&= \left(\frac{m^0(M+1)}{2} + \frac{m^1(M+1)}{2} + ... + \frac{m^{I-1}(M+1)}{2}\right)\\
&= \frac{(M+1)}{2}(m^0 + m^1 + ... + m^{I-1})\\
&= \frac{(M+1)}{2}\frac{1-m^I}{1-m} \qquad (3.3.21)
\end{aligned}$$

Then the expectation of Integration $S_{R,I}$ is given by

$$\mathrm{E}[S_{R,I}] = \frac{(M+1)}{2}\frac{1-m^I}{1-m} \qquad (3.3.22)$$

$\mathrm{Var}[S_{R,I}]$, **variance at intermediate states**

From (2.4.15), the variance at intermediate states is given by $\mathrm{Var}[S_{R,I}] = \mathbf{v}_{R,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{R,I}$. The Variance-covariance matrix $\mathbf{K}_{WW}$ is given by (2.3.8) and modulation vector $\mathbf{v}_{R,I}^{1}$ by (3.3.8). Therefore we have:

$$\begin{aligned}
&\mathrm{Var}[S_{R,I}] \\
&= \mathbf{v}_{R,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{R,I} && (3.3.23)\\
&= (m^0\ m^1\ \ldots\ m^{I-1}\ 0 \ldots 0) \begin{bmatrix} \mathsf{Var}W & \vartheta & \ldots & \vartheta \\ \vartheta & \mathsf{Var}W & \ldots & \vartheta \\ \vdots & \vdots & \ddots & \vdots \\ \vartheta & \vartheta & \ldots & \mathsf{Var}W \end{bmatrix} \begin{bmatrix} m^0 \\ m^1 \\ \vdots \\ m^{I-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} && (3.3.24)\\
&= m^0(m^0\mathsf{Var}W + m^1\vartheta + \ldots + m^I\vartheta) + m^1(m^0\vartheta + m^1\mathsf{Var}W + \ldots + m^I\vartheta) + && (3.3.25)\\
&\ldots + m^{I-1}(m^0\vartheta + m^1\vartheta + \ldots + m^{I-1}\mathsf{Var}W) \\
&= \mathsf{Var}W \sum_{i=0}^{I-1} m^{2i} + && (3.3.26)\\
&\vartheta \left[ m^0 \left( \sum_{i=0}^{I-1} m^i - m^0 \right) + m^1 \left( \sum_{i=0}^{I-1} m^i - m^1 \right) + m^{I-1} \left( \sum_{i=0}^{I-1} m^i - m^{I-1} \right) \right] \\
&= \left( \frac{1-m^{2I}}{1-m^2} \right) \mathsf{Var}W + \left[ \sum_{i=0}^{I-1} m^i \left( \frac{1-m^I}{1-m} - m^i \right) \right] \vartheta && (3.3.27)
\end{aligned}$$

Using the geometric summation formulas, we get:

$$\mathrm{Var}[S_{R,I}] = \left( \frac{1-m^{2I}}{1-m^2} \right) \mathsf{Var}W + \left[ \left( \frac{1-m^I}{1-m} \right)^2 - \frac{1-m^{2I}}{1-m^2} - \right] \mathsf{Cov}(W_i, W_j) \qquad (3.3.28)$$

where Var$W$ is given by (3.2.8) and Cov$(W_i, W_j)$ by (3.2.19).

### 3.3.4 Maximum value of Integration-Final Potential, max($S_{R,M}$)

The maximum value of integration $S_{R,M}$ is defined in equation (2.4.18). we have that $\mathbf{w}^1 = (M, M-1, \ldots, 1)$ and modulation vector is given by (3.3.8). Thus $\max(S_{R,M})$ is

$$\begin{aligned}
&=< \mathbf{w}^1, \mathbf{v}_{R,M}^1 > \\
&= (M-0)m^0 + (M-1)m^1 + (M-2)m^2 + \ldots + (M-(M-1))m^{M-1} \\
&= M(m^0 + m^1 + \ldots + m^{M-1}) - (m^1 + 2m^2 + \ldots + (M-1)m^{M-1}) && (3.3.29)\\
&. = M \sum_{i=0}^{M-1} m^i - \sum_{i=1}^{M-1} i\, m^i && (3.3.30)\\
&= M \left( \frac{1-m^M}{1-m} \right) - \frac{1-m^{M+1}-(Mm^M+1)(1-m)}{(1-m)^2} && (3.3.31)
\end{aligned}$$

Simplifying, the max of integration for ROC is given by:

$$\max(S_{R,M}) = \frac{(1-m)(1+M) - (1-m^{M+1})}{(1-m)^2} \tag{3.3.32}$$

# 4

# $N$-of-$M$ codes-Hypergeometric distribution

For the sake of interest of implementing a sparse distributed memory (SPM), Furber at al. ([11]) raised the point of comparing two $N$-of-$M$ codes differing by $N$, let's say $\mathcal{L}$-of-$M$ and $\mathcal{H}$-of-$M$. They suggest that the probability that two such random $N$-of-$M$ codes share $x$ 1s follows the hypergeometric distribution but they hardly provide a proof.

Traditionally, hypergeometric distribution is associated with the following random experiment: let us consider a population, of size $M$, in which there are exactly $N$ individuals holding a given character ; sample $S$ individuals ; what are the odds that your sample contains $0, \dots, S$ individuals with that character ?

Here, we extend this picture by showing that, more generally, the hypergeometric distribution governs the result of the inner product of two random binary $(0/1)$ vectors of size $M$, once the number of 1s contained in each vector, say $\mathcal{L}$ and $\mathcal{H}$, is fixed for each vector.

The corresponding random experiment would be: sample a binary vector $\mathbf{x} \in \{0,1\}^M$; sample a binary vector $\mathbf{y} \in \{0,1\}^M$; what are the odds that their inner product $X =< \mathbf{x}, \mathbf{y} >$ is $0, \dots, M$ ?

The broader experiment could then be depicted by: let us consider an homogeneous population, of size $M$ ; now, choose a number of token $i \in (1, \dots, M-1)$ to be distributed and give one token to $i$ individuals, choosing them randomly ; then choose a sample size $j \in (1, \dots, M-1)$ and randomly pick $j$ individuals ; what are the odds that your sample contains $0, \dots, M$ individuals which have been given a token ? In that regard, the traditional picture is only a particular case of the broader picture, in the case when the vector representing the individuals that hold the character / have a token is considered as predetermined and fixed, and the sample size has been fixed.

Below, we use linear algebra to prove, in the first section, that the inner product of such vectors $\mathbf{x}, \mathbf{y}$ follows the hypergeometric distribution conditionned by $(M, i, j)$.

In the second section, we will deduce that the first two moments of the final potential of integration random variable for NofM coding are the expected value and variance of an

hypergeometric distribution with parameters $M, \mathcal{L}, \mathcal{H}$.

## 4.1 Preliminaries

### 4.1.1 Definitions

**Definition 4.1.1** (Random Variable $X$)**.** Let $\mathbf{x} \in \{0,1\}^M$ denote a $M \times 1$ random binary vector and $\mathbf{y} \in \{0,1\}^M$ denote a $M \times 1$ random binary vector. The random variable $X$ is defined by the dot product

$$X =< \mathbf{x}, \mathbf{y} >$$

.

**Remark.** *Since the random vectors are defined as binary vectors, they corresponds to a sampling done without replacement. By construction, the dot product directly indicates the number of shared 1s: denoting* $\mathbf{x} = (x_1, \ldots, x_M)$*, and* $\mathbf{y} = (y_1, \ldots, y_M)$*:*

$$< \mathbf{x}, \mathbf{y} >= x_1y_1 + x_2y_2 + ... + x_My_M = \sum_{i=1}^{M} x_iy_i \tag{4.1.1}$$

*yields the number of case where* $x_iy_i$ *trials both received a token and were sampled.*

The following combinatorics formulas will be used throughout the proof:

$${}_nC_r = \binom{n}{r} = \frac{n!}{r!(n-r)!}, \quad n \geq r \tag{4.1.2}$$

$$\binom{n}{1} = n, \quad \binom{n}{0} = \binom{n}{n} = 1, \quad \binom{n}{n-r} = \binom{n}{r} \tag{4.1.3}$$

### 4.1.2 Sample space of $X$

The possible outcomes for all possible combinations of the dot product $< \mathbf{x}, \mathbf{y} >$ for all $(i, j) \in (1, \ldots, M-1) \times (1, \ldots, M-1)$, can condense in a single matrix $\mathbb{A}$. Equivalently, the sample space of the random variable $X$ are the elements of the matrix $\mathbb{A}$. $\mathbb{A}$ is an $2^M \times 2^M$ matrix.

Let's consider matrix $\mathbf{A}$ defined by:

$$\mathbf{A} = [A_{M\times\binom{M}{1}} \quad A_{M\times\binom{M}{2}} \quad ... \quad A_{M\times\binom{M}{M-1}}] \tag{4.1.4}$$

where $A_{M\times\binom{M}{k}}$ contains all possible combinations of $k$ number of ones (1s) in the vector of size $M$, with $k = 1, 2, ..., M-1$.

Then $\mathbf{A}$ is a $M \times 2^M$ matrix and at he same time a $1 \times \binom{M}{M-1}$ block matrix.

Then $\mathbb{A}$ can be partitioned into $\binom{M}{M-1} \times \binom{M}{M-1}$ sub-matrices, considering it as the result of matrix multiplication of $\mathbf{A}$ by its transpose $\mathbf{A}^T$.

Therefore, $\mathbb{A}$ can be written as:

$$\mathbb{A} = \mathbf{A}^T\mathbf{A} \tag{4.1.5}$$

$$= \begin{bmatrix} A_{\binom{M}{1}\times M} \\ A_{\binom{M}{2}\times M} \\ \vdots \\ A_{\binom{M}{M-1}\times M} \end{bmatrix} \begin{bmatrix} A_{M\times\binom{M}{1}} & A_{M\times\binom{M}{2}} & \dots & A_{M\times\binom{M}{M-1}} \end{bmatrix} \tag{4.1.6}$$

$$= \begin{bmatrix} \mathcal{A}_{\binom{M}{1}\times\binom{M}{1}} & \mathcal{A}_{\binom{M}{1}\times\binom{M}{2}} & \cdots & \mathcal{A}_{\binom{M}{1}\times\binom{M}{M-1}} \\ \mathcal{A}_{\binom{M}{2}\times\binom{M}{1}} & \mathcal{A}_{\binom{M}{2}\times\binom{M}{2}} & \cdots & \mathcal{A}_{\binom{M}{2}\times\binom{M}{M-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_{\binom{M}{M-1}\times\binom{M}{1}} & \mathcal{A}_{\binom{M}{M-1}\times\binom{M}{2}} & \cdots & \mathcal{A}_{\binom{M}{M-1}\times\binom{M}{M-1}} \end{bmatrix} \tag{4.1.7}$$

$$= \begin{bmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} & \cdots & \mathbb{A}_{1(M-1)} \\ \mathbb{A}_{21} & \mathbb{A}_{22} & \cdots & \mathbb{A}_{2(M-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{A}_{(M-1)1} & \mathbb{A}_{(M-1)2} & \cdots & \mathbb{A}_{(M-1)(M-1)} \end{bmatrix} \tag{4.1.8}$$

We denote the elements of $\mathbb{A}$ as $\mathbb{A}_{ij}$ which is a $\binom{M}{i}\times\binom{M}{j}$, matrix with $i, j = 1, 2, ..., M-1$. Note that the sub-matrix $\mathbb{A}_{ij}$ is the results of multiplying matrices $A_{\binom{M}{i}\times M}$ that contains all the combinations for a vector of size $M$ containing $i$ 1s and $A_{M\times\binom{M}{j}}$ that contains all the combinations for a vector of size $M$ containing $j$ 1s.

**Remark.** *In terms of distribution of $X$, $\mathbb{A}_{ij}$ is the sample space of $X$ once the couple $(i, j)$ has been sampled.*

**Remark.** *From linear algebra, we know that the product of a matrix with its transpose is a symmetric matrix. Therefore, $\mathbb{A}$ is a $2^M \times 2^M$ symmetric matrix.*

Note that considering $\mathbb{A}$ partitioned into blocks of matrices, that is, the $\binom{M}{M-1}\times\binom{M}{M-1}$ block matrix, $\mathbb{A}$ is still a symmetric matrix since its matrix-elements in the position $ij$th contains the same elements than in the position $ji$th, although $\mathbb{A}_{ij} \neq \mathbb{A}_{ji}$. These matrices do not have the same size but they share same elements because $\mathbb{A}$ is symmetric, and, as far as sample space for $X$ is concerned, the important point is that we have:

$$\mathbb{A}_{ij} = \mathbb{A}_{ji}^T \tag{4.1.9}$$

This fact also stems from the commutative property of the dot product. The entries of $\mathbb{A}$ are the dot product of vectors defined in equation 4.1.1, and this product commutes. If vectors are identified with row or column matrices, these vectors are a row of $\mathbf{A}^T$ and a column of $\mathbf{A}$.

Conversely, from the symmetry of the problem in the hypergeometric distribution, the following identity apply:

$$\frac{\binom{\mathcal{H}}{x}\binom{M-\mathcal{H}}{\mathcal{L}-x}}{\binom{M}{\mathcal{L}}} = \frac{\binom{\mathcal{L}}{x}\binom{M-\mathcal{L}}{\mathcal{H}-x}}{\binom{M}{\mathcal{H}}} \tag{4.1.10}$$

meaning that to draw $\mathcal{L}$ individuals of the population of size $M$ that contains $\mathcal{H}$ successes is just the same as drawing $\mathcal{H}$ individuals from the population $M$ with $\mathcal{L}$ successes.

To prove that $X$ follows the hypergeometric distribution taking advantage of that $\mathbf{A}$ is symmetric, we consider the upper triangular $\mathbb{A}$ block matrix given by:

$$\mathbb{A} = \begin{bmatrix} I & A_{M\times\binom{M}{2}} & \cdots & A_{M\times\binom{M}{M-1}} \\ & \mathbb{A}_{22} & \cdots & \mathbb{A}_{2(M-1)} \\ & & \ddots & \vdots \\ & & & \mathbb{A}_{(M-1)(M-1)} \end{bmatrix} \tag{4.1.11}$$

The first row is composed by the matrices shown because for $i = 1$, the matrix $A_{\binom{M}{1}\times M}$ is the identity matrix $I$ which contains all possible combinations to arrange one $1$ in a vector of size $M$. Then for $i = 1$ and $j = 1, ..., M-1$

$$\mathbb{A}_{1j} = A_{\binom{M}{1}\times M} A_{M\times\binom{M}{j}} = A_{M\times\binom{M}{j}} \tag{4.1.12}$$

$\mathbb{A}$ contains all possible outcomes of the dot product $< \mathbf{x}, \mathbf{y} >$ for $i$ and $j$. The sample space of the experiment $\mathbb{A}_{ij}$ consists of $\binom{M}{i} \times \binom{M}{j}$ possible outcomes that are equally likely, that is, the probability for each outcome is:

$$\frac{1}{\binom{M}{i} \times \binom{M}{j}}$$

Having the sample space well defined, the proof consists in finding the probability of the outcomes and showing that it is hypergeometric.

### 4.1.3 Tree-method to build a matrix combinations of $\binom{M}{k}$

All possible $M$ of $k$ combinations will build in a matrix by columns. We start by defining matrix notation.

**Definition 4.1.2.** $\binom{M}{i}^*$ matrix represents the transpose of the matrix $A_{\binom{M}{i}\times M}$. Let $\binom{\mathcal{M}}{k}_1$ be the number of $1'$s and $\binom{\mathcal{M}}{k}_0$ the number of $0'$s of the matrix $\binom{M}{i}^*$.

**Definition 4.1.3** (Associated matrix)**.** $[\binom{\mathcal{M}}{k}]$ denote a $\mathcal{M} \times \binom{\mathcal{M}}{k}$ matrix that represents all the combinations $\mathcal{M}$ choose $k$. For $k = 0$, $[\binom{\mathcal{M}}{0}]$ is a $M \times 1$ column vector of zeros, and for $k = \mathcal{M}$, $[\binom{\mathcal{M}}{\mathcal{M}}]$ is a $\mathcal{M} \times 1$ column vector of ones.

We start to fill the matrix $\binom{M}{i}^*$ with ones and zeros by rows where the number of zeros and ones is determined by the columns of the associated sub-matrix defined in 4.1.3. To get the first row we have two associated matrices one for the $1'$s and the other for the $0'$s. The associated matrix for ones is equivalent to fixing an one in the initial or base matrix and calculating the combinations that are obtained, so we have a new matrix with one less one $(1)$ and one less element of the base matrix.

Therefore, the parameters are $\mathcal{M} = M - 1$ considering that there is one less element of the vector and $k_1 = i - 1$ since that there is one $(1)$ less. Equivalently, for the associated

matrix for zeros, the parameters are $\mathcal{M} = M - 1$ because there is one less element of the vector and $k_2 = i$ since that the ones $1'$s keep constant. The matrix is given by,

$$\binom{M}{i}^* = \left(\begin{array}{c|c} 1'\text{s} & 0's \\ \hline [\binom{\mathcal{M}}{k_1}] & [\binom{\mathcal{M}}{k_2}] \end{array}\right)_{M\times\binom{M}{i}} = \left(\begin{array}{c|c} \binom{M-1}{i-1}_1 & \binom{M-1}{i}_0 \\ \hline [\binom{M-1}{i-1}] & [\binom{M-1}{i}] \end{array}\right)_{M\times\binom{M}{i}} \tag{4.1.13}$$

note that $\binom{M-1}{i-1}_1 + \binom{M-1}{i}_0 = \binom{M}{i}$. For the second row we have four associated sub-matrices with common parameter $\mathcal{M} = M - 2$ and $k_3 = i - 2, k_4 = i - 1, k_5 = i - 1, k_6 = i$ and so on. It is important to highlight the fact that the value of $k$ by row decreases by one. Thus we have the following matrix,

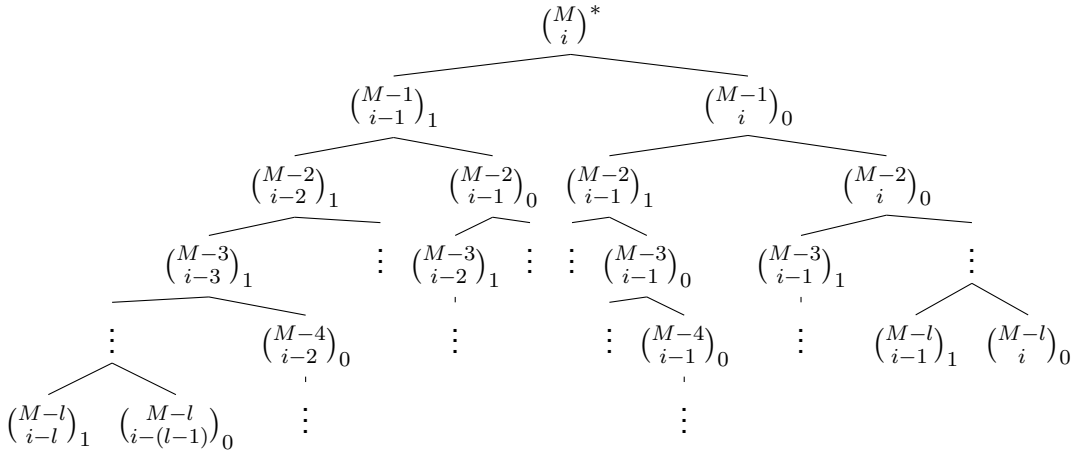
$$\binom{M}{i}^* = \left(\begin{array}{cc|cc} \multicolumn{2}{c|}{\text{1's}} & \multicolumn{2}{c}{\text{0's}} \\ \hline \text{1's} & \text{0's} & \text{1's} & \text{0's} \\ \hline [\binom{\mathcal{M}}{k_3}] & [\binom{\mathcal{M}}{k_4}] & [\binom{\mathcal{M}}{k_5}] & [\binom{\mathcal{M}}{k_6}] \end{array}\right)_{M\times\binom{M}{i}} \tag{4.1.14}$$

$$= \left(\begin{array}{cc|cc} \multicolumn{2}{c|}{\binom{M-1}{i-1}_1} & \multicolumn{2}{c}{\binom{M-1}{i}_0} \\ \hline \binom{M-2}{i-2}_1 & \binom{M-2}{i-1}_0 & \binom{M-2}{i-1}_1 & \binom{M-2}{i}_0 \\ \hline [\binom{M-2}{i-2}] & [\binom{M-2}{i-1}] & [\binom{M-2}{i-1}] & [\binom{M-2}{i}] \end{array}\right)_{M\times\binom{M}{i}} \tag{4.1.15}$$

For example, $\binom{6}{3}^*$ matrix can be obtained like this:

$$\binom{6}{3}^* = \begin{bmatrix} & & \binom{5}{2}_1 & & & & \binom{5}{3}_0 & \\ & \binom{4}{1}_1 & & \binom{4}{2}_0 & & \binom{4}{2}_1 & & \binom{4}{3}_0 \\ \binom{3}{0}_1 & \binom{3}{1}_0 & \binom{3}{1}_1 & \binom{3}{2}_0 & \binom{3}{1}_1 & \binom{3}{2}_0 & \binom{3}{2}_1 & \binom{3}{3}_0 \\ [\binom{3}{0}] & [\binom{3}{1}] & [\binom{3}{1}] & [\binom{3}{2}] & [\binom{3}{1}] & [\binom{3}{2}] & [\binom{3}{2}] & [\binom{3}{3}] \end{bmatrix} \tag{4.1.16}$$

We represent the matrix $\binom{M}{i}^*$ as a tree, where each line $l$ represents the $i$th row of the matrix, then we have the following representation:




*Tree-matrix* (4.1.17)

For $l = i$, we get the last row of sub-matrices, then for definition 4.1.3, the first sub-matrix $[\binom{M-i}{0}]$ is a $1 \times (M-i)$ vector of zeros. $M$ takes values greater than $i$, $M > i$ then we have

combinatorics formula that are not defined since that for a $\binom{M}{k}$, $M < k$. This happening for a $i$ fixed and $M$ taking values $i+1, \cdots, i+l-1$ since that the last combinatorics defined is when $M - l = i$.

Another useful representation is denoting the combinatorics in the tree-matrix as $\binom{M-l}{i-r}$ where $r$ takes the values $l, l-1, \cdots, 1, 0$. Note that for each row $l$,in 5.2.8, the only variable is $r$. We consider only $r$ in a table-matrix and we color the cells of ones.

| 1 | | | | 0 | | | |
|---|---|---|---|---|---|---|---|
| 2 | | 1 | | 1 | | 0 | |
| 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| 4 3 | 3 2 | 3 2 | 2 1 | 3 1 | 2 1 | 2 1 | 1 0 |
| ⋮ | | ⋮ | | ⋮ | | ⋮ | |

(4.1.18)

**Remark.** *The number $r$ coincides with the number of ones that we have in each set of columns, that is, the gray blocks. This is because the combinatorics of ones are always increased once. The number of columns is the columns of the associated matrix for the gray block. For example, $r = 4$ is $\binom{M-l}{i-4}_1$, the matrix associated is $[\binom{M-l}{i-4}]$*

## 4.2 $N$-of-$M$ match probabilities

**Theorem 4.2.1.** *For a given $M$, the inner product $< \mathbf{x}, \mathbf{y} >$ defined by 4.1.1, follows the hypergeometric distribution, once $(i, j)$ is given.*

*Proof.* Let us divide the proof into three cases,

### Case I : $i = 1$ and $j = 1, ..., M-1$.

We consider the matrices on the first row of the $\mathbb{A}$ block matrix,

$$\mathbb{A}_{1j} = \begin{bmatrix} I & A_{M\times\binom{M}{2}} & \cdots & A_{M\times\binom{M}{M-1}} \end{bmatrix} \tag{4.2.1}$$

The possible outcomes of these matrices are $\{0, 1\}$ because equation 4.1.12 and the number of outcomes is the size of the matrices $M \times \binom{M}{M-j}$. Each matrix has $j$ ones (1s) and $M - j$ zeros. Then the total amount of 1s, is $j$ multiplied by the number of columns, $\binom{M}{M-j}$. The frequency of one is $f_1 = j\binom{M}{M-j}$. In the same way, we have that the frequency of zero $f_0 = (M-j)\binom{M}{M-j}$. As each outcome occurs with the same probability, then the likelihood of any event becomes:

$$P(X=0) = \frac{f_0}{M \times \binom{M}{M-j}} = \frac{(M-j)\binom{M}{M-j}}{M \times \binom{M}{M-j}} \tag{4.2.2}$$

$$= \frac{(M-j)}{M} = \frac{\binom{M-j}{1}}{\binom{M}{1}} = \frac{\binom{j}{0}\binom{M-j}{1-0}}{\binom{M}{1}} \tag{4.2.3}$$

which is the probability of the hypergeometric random variable $X = 0$ with $i = 1$ individual drawn of the population $M$ and $j$ successes in the population.

Similarly, the probability for $X = 1$ is given by,

$$P(X = 1) = \frac{f_1}{M \times \binom{M}{M-j}} = \frac{j\binom{M}{M-j}}{M \times \binom{M}{M-j}} \tag{4.2.4}$$

$$= \frac{j}{M} = \frac{\binom{j}{1}\binom{M-j}{1-1}}{\binom{M}{1}} \tag{4.2.5}$$

which is the probability of the hypergeometric random variable $X = 1$ with $i = 1$ individual drawn of the population $M$ and $j$ successes in the population.

We have then the following formulas for the probability:

$$P(X = 0) = \frac{M - j}{M}, P(X = 1) = \frac{j}{M}$$

If we define the probability as functions depending of $M$ and fixed $j$, we would have that:

$$P_{0,j}(M) = \frac{M - j}{M}, P_{1,j}(M) = \frac{j}{M}$$

so that, when $M \to \infty$ then $P_0 \to 1$ and $P_1 \to 0$.

**Case II :** $i = 1, ..., M - 1$, **and** $j = M - 1$.

This case considers the matrices of the last column of the $\mathbb{A}$ block matrix,

$$\mathbb{A}_{i(M-1)} = \begin{bmatrix} A_{M\times\binom{M}{M-1}} \\ \mathbb{A}_{2\ (M-1)} \\ \vdots \\ \mathbb{A}_{(M-1)\ (M-1)} \end{bmatrix} \tag{4.2.6}$$

We have that $\mathbb{A}_{i(M-1)} \equiv \mathbb{A}_{(M-1)i}$ from equation 4.1.9, then we use the $\mathbb{A}_{(M-1)i}$ matrix to find the possible outcomes and its frequency.

In linear algebra, the product $C$ of two matrices $A$ and$B$ can be writing as the result of taking linear combinations of the columns of $A$ with the scalars given by columns of $B$. We use this representation for the matrix $\mathbb{A}_{(M-1)i}$ since it is defined by the product of the matrices for $i = 1, ..., M - 1$,

$$\mathbb{A}_{(M-1)i} = A_{\binom{M}{M-1}\times M} A_{M\times\binom{M}{i}} \tag{4.2.7}$$

We solve this product by representing it as linear combination of the columns of $A_{\binom{M}{M-1}\times M}$ with scalars given by columns of $A_{M\times\binom{M}{i}}$.

Let us denote the columns of matrix $A_{\binom{M}{M-1}\times M}$ by $\mathbf{a}^q$, and the columns of matrix $A_{M\times\binom{M}{i}}$ by $\mathbf{b}^k$. The $k$th column is denoted by $(b_{1k}\ b_{2k} \cdots b_{Mk})^T$ for $k = 1, ..., \binom{M}{i}$. Thus, the $k$th column of the matrix $\mathbb{A}_{(M-1)i}$ denote by $\mathbb{A}^k_{(M-1)i}$, is given by,

$$\mathbb{A}^k_{(M-1)i} = b_{1k}\mathbf{a}^1 + b_{2k}\mathbf{a}^2 + \cdots + b_{Mk}\mathbf{a}^M \tag{4.2.8}$$

The coefficients $b_{1k}, b_{2k}, \cdots, b_{Mk}$ are ones or zeros. It means that since each column $\mathbf{b}^k$ contains $i$ ones (1s) and $M-i$ zeros, we add up $i$ different columns in each $k$th combination, but with the characteristic that each column contains the same number of ones, which is $M-1$.

The $k=1$ combination, that is, the first column of the matrix $\mathbb{A}_{(M-1)i}$ is given by:

$$\mathbb{A}^1_{(M-1)i} = \mathbf{a}^1 + \mathbf{a}^2 + \cdots + \mathbf{a}^i = \begin{bmatrix}1\\1\\\vdots\\1\\0\end{bmatrix} + \begin{bmatrix}1\\1\\\vdots\\0\\1\end{bmatrix} + \cdots + \begin{bmatrix}0\\1\\\vdots\\1\\1\end{bmatrix} = \begin{pmatrix}i-1\\i\\\vdots\\i-1\\i-1\end{pmatrix} \tag{4.2.9}$$

Note that, if there is a zero in a row, we add up $i-1$ ones and if there is not zero, we add up $i$ ones (1s), which means that the possible outcomes are $\{i-1, i\}$ for all columns of the matrix.

Now we calculate the frequency of the outputs. We get $i$ times the output $i-1$, then as the size of each column is $M \times 1$ then $M-i$ is the times for $i$. Multiplying for all columns, we get the frequency for each outcome,

$$f_{i-1} = i\binom{M}{i}, \quad f_i = (M-i)\binom{M}{i} \tag{4.2.10}$$

given that $\mathbb{A}_{(M-1)i}$ is a $\binom{M}{M-1} \times \binom{M}{i}$ matrix then the sample space contains $\binom{M}{M-1} \times \binom{M}{i}$ outcomes.

Therefore, the probabilities are given by,

$$P(X = i-1) = \frac{f_{i-1}}{M \times \binom{M}{i}} = \frac{i\binom{M}{i}}{M \times \binom{M}{i}} = \frac{i}{M} \tag{4.2.11}$$

$$P(X = i) = \frac{f_i}{M \times \binom{M}{i}} = \frac{(M-i)\binom{M}{i}}{M \times \binom{M}{i}} = \frac{M-i}{M} \tag{4.2.12}$$

The probability of the random variable $X = i$ for $i = 1, ..., M-1$ individuals drawn of the population and $j = M-1$, the successes in $M$ is given by,

$$P(x = i) = \frac{\binom{M-1}{i}\binom{M-(M-1)}{i-i}}{\binom{M}{i}} = \frac{\binom{M-1}{i}\binom{1}{0}}{\binom{M}{i}} = \frac{\binom{M-1}{i}}{\binom{M}{i}} = \frac{M-i}{M} \tag{4.2.13}$$

$$P(x = i-1) = \frac{\binom{M-1}{i-1}\binom{M-(M-1)}{i-(i-1)}}{\binom{M}{i}} = \frac{\binom{M-1}{i-1}\binom{1}{1}}{\binom{M}{i}} = \frac{\binom{M-1}{i-1}}{\binom{M}{i}} = \frac{i}{M} \tag{4.2.14}$$

Using the combinatorial properties and simplifying, we get the same probability. Defining the probability as functions depending of $M$, we would have:

$$P_{i-1}(M) = \frac{i}{M}, P_i(M) = \frac{M-i}{M}$$

so that, when $M \to \infty$, then $P_{i-1} \to 0$ and $P_i \to 1$.

**Case III :** $i = 2, ..., M-1$, **and** $j = 2, ..., M-1$.

So far, it has been proved that $X$ follows the hypergeometric distribution for $i = 1$ with $j = 1, ..., M-1$ and for $j = M-1$ with $i = 1, ..., M-1$, that is, the following set of matrices:

$$\mathbb{A}_{1j} = A_{\binom{M}{1}\times M} A_{M\times\binom{M}{j}} \tag{4.2.15}$$

$$\mathbb{A}_{i(M-1)} = A_{\binom{M}{i}\times M} A_{M\times\binom{M}{M-1}} \tag{4.2.16}$$

What was done to obtain the outcomes and its frequency of the previous cases were to identify patterns and count. We deduce a formula that gives the outcomes of each experiment $\mathbb{A}_{ij}$, for $j = 1, ..., M-1$, and a general method to find the frequency of the following matrices

$$\mathbb{A}_{2j} = A_{\binom{M}{2}\times M} A_{M\times\binom{M}{j}} \tag{4.2.17}$$

$$\mathbb{A}_{3j} = A_{\binom{M}{3}\times M} A_{M\times\binom{M}{j}} \tag{4.2.18}$$

$$\vdots \quad = \quad \vdots$$

$$\mathbb{A}_{ij} = A_{\binom{M}{i}\times M} A_{M\times\binom{M}{j}} \tag{4.2.19}$$

We start by finding the outputs and its frequency for each experiment $\mathbb{A}_{ij}$, which means that we must obtain the product of two matrices. The matrix multiplication will be represented by linear combination of columns of $\mathbf{A}^T$ in the same way as case **II**. From this representation it should be observed that *to obtain the outputs and frequency of the experiment* $\mathbb{A}_{ij}$ *it is enough to know the first column from the matrix product:*

$$\mathbb{A}_{ij} = A_{\binom{M}{i}\times M} A_{M\times\binom{M}{j}} \tag{4.2.20}$$

*and it is obtained only by adding up the first* $j$ *columns of the matrix* $A_{\binom{M}{i}\times M}$ (See equation 4.2.9). Thus, from here on, we only consider the matrix $A_{\binom{M}{i}\times M}$ and use tree-matrix representation $\binom{M}{i}^* = (A_{\binom{M}{i}\times M})^T$ defined in the subsection 4.1.3. For it representation, instead of adding $j$ columns we add $j$ rows to get the outputs of the experiment $\mathbb{A}_{ij}$. We consider the experiments $i \geq j$, since the symmetry of the problem, see equation 4.1.9.

Using the notation of tree-matrix in 5.2.8 for the matrix $\binom{M}{i}^*$, we add up the $j$th row, then the last row of zeros and ones and its associated matrix are given by:

$$\binom{M-j}{i-j}_1 \quad \binom{M-j}{i-(j-1)}_0 \quad \cdots \quad \binom{M-j}{i-1}_1 \quad \binom{M-j}{i}_0 \tag{4.2.21}$$

$$\left[\binom{M-j}{i-j}\right] \quad \left[\binom{M-j}{i-(j-1)}\right] \quad \cdots \quad \left[\binom{M-j}{i-1}\right] \quad \left[\binom{M-j}{i}\right] \tag{4.2.22}$$

depending on the value of $M$ there are combinatorics that are not defined. The minimum value that $M$ can take is $i+1$ so note that the first combinatorics $\binom{M-j}{i-j}_1$ that is ones is always defined for all $M \geq i+1$. *This implies that we always have as output* $x = j$ *which would be the maximum value. The other possible outputs decrease by*

*one, so $x = j, j-1, \cdots$, but the existence of the other outputs depends on whether the combinatorics is defined (subsection 4.1.3 ).*

We get the frequency for each output by the associated matrix defined in 4.1.3 and the construction of a table representation 4.1.18. *So we know the times the output is repeated in a gray block with the number of columns of the associated matrix and now we count how many blocks we have for each output (see remark 4.1.3 ).*

Therefore we have the outputs and frequency of the first column of the experiment $\mathbb{A}_{ij}$. For the other columns we have the same outputs and frequency but in a different order, so it only remains to multiply by the number of columns which is $\binom{M}{j}$ to obtain the probability distribution for this experiment.

Let us find the outputs and frequency for the experiment $\mathbb{A}_{i3}$, then we add up $3$ rows of the matrix $\binom{M}{i}^*$ (see equation 4.2.20,) and for equation 4.2.21 we have the following combinatorics:

$$\binom{M-3}{i-3}_1 \quad \binom{M-3}{i-2}_0 \quad \binom{M-3}{i-2}_1 \quad \binom{M-3}{i-1}_0 \quad \binom{M-3}{i-2}_1 \quad \binom{M-3}{i-1}_0 \quad \binom{M-3}{i-1}_1 \quad \binom{M-3}{i}_0 \tag{4.2.23}$$

from here $x = 3$ is the max output, we could have $x = 2, 1, 0$ but it is determined for the value of $M$. Thus, for the minimal values for $i$ and $M$, $i = 3$ and $M = 4$ we have the combinatorics for the row $l = 3$ and at the same time they are the columns of the associated matrix:

$$\begin{matrix} \binom{1}{0} & \binom{1}{1} & \binom{1}{1} & \binom{1}{2} & \binom{1}{1} & \binom{1}{2} & \binom{1}{2} & \binom{1}{3} \\ & & & \times & & \times & \times & \times \end{matrix} \tag{4.2.24}$$

$\binom{1}{2}$ and $\binom{1}{3}$ are not defined (see eq 4.1.2). It means the outputs are $x = 3, 2$. since that the outputs decrease by one and it correspond to a combinatoric. Thus the columns of the associated matrix for $x = 3$ is $\binom{1}{0}$ and for $x = 2$ is $\binom{1}{1}$ which is repeated $3$ times. We can write this amounts as combinatory so $1 = \binom{3}{3}, 3 = \binom{3}{2}$ it is the frequency for the first column. To get the frequency for the experiment we multiply for the numbers of columns which is $\binom{4}{3}$.

$$f_3 = 1 \binom{1}{0} \binom{4}{3} = \binom{3}{3} \binom{1}{0} \binom{4}{3} \tag{4.2.25}$$

$$f_2 = 3 \binom{1}{1} \binom{4}{3} = \binom{3}{2} \binom{1}{1} \binom{4}{3} \tag{4.2.26}$$

in the same way, for $M = 5$:

$$\begin{matrix} \binom{2}{0} & \binom{2}{1} & \binom{2}{1} & \binom{2}{2} & \binom{2}{1} & \binom{2}{2} & \binom{2}{2} & \binom{2}{3} \\ & & & & & & & \times \end{matrix} \tag{4.2.27}$$

it cancels out the combinatory $\binom{2}{3}$ then outputs are $x = 3, 2, 1$ and its frequency are

$$f_3 = 1 \binom{2}{0} \binom{4}{3} = \binom{3}{3} \binom{2}{0} \binom{4}{3} \tag{4.2.28}$$

$$f_2 = 3 \binom{2}{1} \binom{4}{3} = \binom{3}{2} \binom{2}{1} \binom{4}{3} \tag{4.2.29}$$

$$f_1 = 3 \binom{2}{2} \binom{4}{3} = \binom{3}{1} \binom{2}{2} \binom{4}{3} \tag{4.2.30}$$

$M = 6$

$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{2} \quad \binom{3}{3} \tag{4.2.31}$$

it does not cancel any combinatory, then the outputs are $x = 3, 2, 1, 0$

$$f_3 = 1 \binom{3}{0} \binom{4}{3} = \binom{3}{3} \binom{3}{0} \binom{4}{3} \tag{4.2.32}$$

$$f_2 = 3 \binom{3}{1} \binom{4}{3} = \binom{3}{2} \binom{3}{1} \binom{4}{3} \tag{4.2.33}$$

$$f_1 = 3 \binom{3}{2} \binom{4}{3} = \binom{3}{1} \binom{3}{2} \binom{4}{3} \tag{4.2.34}$$

$$f_0 = 1 \binom{3}{3} \binom{4}{3} = \binom{3}{0} \binom{3}{3} \binom{4}{3} \tag{4.2.35}$$

In general, we have that:

| $k = i - r$ | output$= x$ | frequency | | |
|---|---|---|---|---|
| $i - j$ | $j$ | $\binom{j}{j}$ | $\binom{M-j}{i-j}$ | |
| $i - (j-1)$ | $j-1$ | $\binom{j}{j-1}$ | $\binom{M-j}{i-(j-1)}$ | |
| $i - (j-2)$ | $j-2$ | $\binom{j}{j-2}$ | $\binom{M-j}{i-(j-2)}$ | $\binom{M}{j}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | |
| $i$ | $0$ | $\binom{j}{0}$ | $\binom{M-j}{i}$ | |

therefore, to find the probability, we divide by the total of outputs $\binom{M}{i}\binom{M}{j}$ of the experiment, then

$$p(X = x; M, i, j) = \frac{\binom{j}{x}\binom{M-j}{i-x}\binom{M}{j}}{\binom{M}{i}\binom{M}{j}} = \frac{\binom{j}{x}\binom{M-j}{i-x}}{\binom{M}{i}} \tag{4.2.36}$$

which it is the probability given by hypergeometric distribution with parameters, $M, i, j \in \{1, \cdots, M-1\}$.

□

## 4.3 Integration of NofM coding distributed hypergeometrically with parameters, $M, \mathcal{L}, \mathcal{H}$.

In the chaper of NofM coding we get the formula for the two first moments of the integration, the expected value $\mathrm{E}[S_{F,\mathcal{L}}]$ and the variance $\mathrm{Var}[S_{F,\mathcal{L}}]$. We will proof that the two first moments for the NofM code are the expected value and the variance of the hypergeometric distribution. Thus the expectation of Integration $S_{F,\mathcal{L}}$ is given by

$$\mathrm{E}[S_{F,\mathcal{L}}] = \frac{\mathcal{L}\,\mathcal{H}}{M} \tag{4.3.1}$$

which is the expected value of hypergeometric probability distribution.

Let us prove that the variance of the integration of NofM coding $\mathrm{Var}[S_{F,\mathcal{L}}]$ is the variance of the share $x$ 1s random variable which follows the hypergeometric distribution. Then we have that,

$$\mathrm{Var}[S_{F,\mathcal{L}}] = \mathcal{L}\ \mathsf{Var}(W) + \mathcal{L}\ (\mathcal{L}-1)\ \mathsf{Cov}(W_i, W_j) \tag{4.3.2}$$

$$= \mathcal{L}(\mu_W(1-\mu_W)) + \mathcal{L}\ (\mathcal{L}-1)\left(\mu_W \frac{\mathcal{H}-1}{M-1} - \mu_W^2\right) \tag{4.3.3}$$

$$= N\ \mu_W[1-\mu_W + (\mathcal{L}-1)\frac{\mathcal{H}-1}{M-1} - (\mathcal{L}-1)\mu_W] \tag{4.3.4}$$

$$= \mathcal{L}\ \mu_W\left[1 - \frac{\mathcal{H}\ \mathcal{L}}{M} + \frac{(\mathcal{L}-1)(\mathcal{H}-1)}{M-1}\right] \tag{4.3.5}$$

$$= \frac{\mathcal{H}\ \mathcal{L}}{M}\left[\frac{(M-\mathcal{L})(M-\mathcal{H})}{M(M-1)}\right] \tag{4.3.6}$$

which is the variance of hypergeometric probability distribution.

# 5
# $N$-of-$M$ Coding

NofM (NoM) coding is a TTFS proposal ([**Thorpe2019**, 11]), in which only the $N$ first spikes among $M$ afferents are propagated and these "first spike patterns" can be readout by downstream neurons with binary weights and no desensitization: as a result, the exact order between the first spikes does not matter.

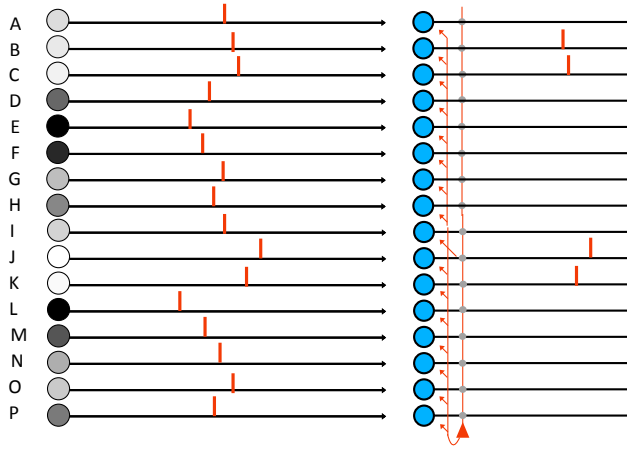



Figure 5.1 – NofM coding with $M = 16$ and $N = 4$. All the afferents fire exactly one spike. This is an binary pattern $\mathbf{x} \in \{0,1\}^{16}$ in which $\mathbf{x}$ is $4$-of$-16$ code, i.e., $4$ bits are on, from the total of $16$. The modulation could be implemented with (k-WTA) inhibitory circuit.

For two random NofM codes, $\mathcal{N}$ of M and $\mathcal{W}$ of M, the probability distribution that two such random NofM codes share $x1s$ (final potential), follows the hypergeometric distribution with $\mathcal{N}$ draws from a population of size $M$ containing $\mathcal{W}$ successes – or, equivalently, $\mathcal{W}$ draws from a population of size $M$ containing $\mathcal{N}$ successes [11].

We perform a proof of this property in the previous chapter(Chapter VI) where we introduce a tree-matrix representation (Chapter VI 4.1.3) which was the main tool to prove

it. It representation will be the key in this chapter to find the bivariate distribution of the NofM weights to obtain the covariance.

Therefore, in this chapter we apply the TTFS coding theory (chapter IV) for NofM to establish the expectation and the variance for the integration at final potential and intermediate states. There are two stages, the first to develop the probabilistic theory of synaptic weights defined as random vectors $\mathbf{W}$ which we call **NofM scores**. In section 5.1 we define the support of the scores $\Omega_F$ to find the joint $P_{W_1W_2\ldots W_M}$ and the marginal $P_{W_i}(w)$ probability mass functions. To obtain the NofM scores where for an input binary pattern $\mathcal{N}$ bits are on from the total of $M$, we use the max function defined in Ranked-NoM, which perform this from the weights and then we remove the order by the indicator function $\mathbf{1}_A$.

In the section 5.2 we establish the expectation vector $\mathrm{E}[\mathbf{W}]$ and the variance-covariance matrix $\mathbf{K}^F_{WW}$. Since the random variables $W_1, W_2, \ldots, W_M$ are not independent we find the covariance $\vartheta$ using a tree method to calculate the frequency to bivariate probability distribution $f(W_i, W_j)$.

The second stage is to develop the statistical theory for the random variable integration $S_{F,I}$ at final $I = \mathcal{N}$ and intermediate states $I < \mathcal{N}$ which consists in computing scores vectors with modulation vector $\mathbf{v}^l_F$ by the inner product. Finally we define a formula for the maximum integration value, $\max(S_{F,\mathcal{N}})$, which will use to compare the codes by discriminability power.

## 5.1 NofM scores-Support $\Omega_F$

### 5.1.1 Scores vector function, $\Phi_F$

The support $\Omega_F$ is generated by using the function of Ranked-NoM coding $\Phi_H$. For NofM, $\Phi_F$ is a composition of indicator function $\mathbf{1}_A$ with the Ranked-NoM function $\Phi_H$ where $A = \{w^k_i \in \Omega_H : w^k_i = \max(0, \hat{w}^k_i - M + \mathcal{W}) \neq 0\}$.

The scores are obtained by the vector-value function $\Phi_F(\hat{\mathbf{w}}^k) = \mathbf{w}^l_F$ defined in (2.2.3). Then the vectorial components are given by:

$$\phi_{F,i}(\hat{\mathbf{w}}^k) = \phi_{F,i}(\hat{w}^k_1, \ldots, \hat{w}^k_M) = \mathbf{1}_A \circ \phi_{H,i}(\hat{w}^k_1, \ldots, \hat{w}^k_M), \tag{5.1.1}$$

such as

$$\begin{aligned}
\Phi_F(\hat{\mathbf{w}}^k) &= \Phi_F(\hat{w}^k_1, \ldots, \hat{w}^k_M) && (5.1.2)\\
&= (\phi_{F,1}(\hat{w}^k_1, \ldots, \hat{w}^k_M), \ldots, \phi_{F,M}(\hat{w}^k_1, \ldots, \hat{w}^k_M)) && (5.1.3)\\
&= (\mathbf{1}_A(\phi_{H,1}(\hat{w}^k_1, \ldots, \hat{w}^k_M)), \ldots, \mathbf{1}_A(\phi_{H,M}(\hat{w}^k_1, \ldots, \hat{w}^k_M))) && (5.1.4)\\
&= (\mathbf{1}_A(\max(0, \hat{w}^k_i - M + \mathcal{W}), \ldots, (\mathbf{1}_A(\max(0, \hat{w}^k_i - M + \mathcal{W})))) && (5.1.5)\\
&= (w^l_1, \ldots, w^l_M) = \mathbf{w}^l_F && (5.1.6)
\end{aligned}$$

Note that for $k = 1$, we have that $\hat{\mathbf{w}}^1 = (M, M-1 \ldots, 1)$, thus

$$\Phi_F(\hat{\mathbf{w}}^1) = (\mathbf{1}_A(\phi_{H,1}(\hat{w}_1^1, \ldots, \hat{w}_M^1)), \ldots, \mathbf{1}_A(\phi_{H,M}(\hat{w}_1^1, \ldots, \hat{w}_M^1))) \tag{5.1.7}$$
$$= (\mathbf{1}_A(\max(0, \mathcal{W})), \ldots, \mathbf{1}_A(\max(0, \mathcal{W} - M + 1))) \tag{5.1.8}$$
$$= (\mathbf{1}_A(\mathcal{W}), \mathbf{1}_A(\mathcal{W} - 1), \ldots, \mathbf{1}_A(0)) \tag{5.1.9}$$
$$= (1, 1, \ldots, 0) = \mathbf{w}_F^1 \tag{5.1.10}$$

The vectors in the support of Ranked-NoM coding by the function $\Phi_F$ get the vectors in $\Omega_H$ converted into vectors of ones and zeros. Therefore, we have that $\Omega_F$ gets reduced vectors each time we generated vectors from $\Omega_H$, because we aren't interested in their order any more. Then we divide by the number of ways that you can arrange $\mathcal{W}$ numbers, which is $\mathcal{W}!$. Thus the cardinality of support of NofM, $\Omega_F$ is:

$$\frac{|\Omega_H|}{\mathcal{W}!} = \frac{M!}{\mathcal{W}!(M-\mathcal{W})!} = \binom{M}{\mathcal{W}} = |\Omega_F| \tag{5.1.11}$$

thus $\Omega_F = \{\mathbf{w}^1, ..., \mathbf{w}^{|\Omega_F|}\}$.

**Example 5.1.1.** As an illustration, let $M = 4$ and $\mathcal{W} = 2$ ($|\Omega| = 24$, $|\Omega_H| = 12$ permutations and $|\Omega_F| = \binom{4}{2} = 6$ combinations.). We would have, for the best order:

$$\Phi_F(\hat{\mathbf{w}}^1) = (\mathbf{1}_A(\max(0, 2)), \mathbf{1}_A(\max(0, 1)), \mathbf{1}_A(\max(0, 0)), \mathbf{1}_A(\max(0, -1))) \tag{5.1.12}$$
$$= (\mathbf{1}_A(2), \mathbf{1}_A(1), \mathbf{1}_A(0), \mathbf{1}_A(0))$$
$$= (1, 1, 0, 0) = \mathbf{w}_H^1 \tag{5.1.13}$$

and for the worst order:

$$\Phi_F(\hat{\mathbf{w}}^{24}) = (\mathbf{1}_A(\max(0, -1)), \mathbf{1}_A(\max(0, 0)), \mathbf{1}_A(\max(0, 1)), \mathbf{1}_A(\max(0, 2))) \tag{5.1.14}$$
$$= (\mathbf{1}_A(0), \mathbf{1}_A(0), \mathbf{1}_A(1), \mathbf{1}_A(2))$$
$$= (0, 0, 1, 1) = \mathbf{w}_H^6 \tag{5.1.15}$$

Note that for $\hat{\mathbf{w}}^{18} = (2, 1, 3, 4)$ and $\hat{\mathbf{w}}^{23} = (1, 2, 4, 3)$ we would also obtain:

$$\Phi_F(\hat{\mathbf{w}}^{18}) = (\mathbf{1}_A(\max(0, 0)), \mathbf{1}_A(\max(0, -1)), \mathbf{1}_A(\max(0, 1)), \mathbf{1}_A(\max(0, 2))) \tag{5.1.16}$$
$$= (\mathbf{1}_A(0), \mathbf{1}_A(0), \mathbf{1}_A(1), \mathbf{1}_A(2))$$
$$= (0, 0, 1, 1) = \mathbf{w}_H^6 \tag{5.1.17}$$
$$\Phi_F(\hat{\mathbf{w}}^{23}) = (\mathbf{1}_A(\max(0, -1)), \mathbf{1}_A(\max(0, 0)), \mathbf{1}_A(\max(0, 2)), \mathbf{1}_A(\max(0, 1))) \tag{5.1.18}$$
$$= (\mathbf{1}_A(0), \mathbf{1}_A(0), \mathbf{1}_A(2), \mathbf{1}_A(1))$$
$$= (0, 0, 1, 1) = \mathbf{w}_H^6 \tag{5.1.19}$$

therefore, $\Phi_F(\hat{\mathbf{w}}^{24}) = \Phi_F(\hat{\mathbf{w}}^{18}) = \Phi_F(\hat{\mathbf{w}}^{23}) = \mathbf{w}_H^6$.

### 5.1.2 Probability distributions functions

Let $\mathbf{W} = (W_1\ W_2\ \ldots\ W_M)^\mathsf{T}$ be a discrete random vector with support the ordered set $\Omega_F = \{\mathbf{w}^1, \ldots, \mathbf{w}^{\binom{M}{\mathcal{W}}}\}$, which is generated by the vector-function $\Phi_F$ defined in (5.1.1),

thus the support is given by:

$$\Omega_F = \Omega_{W_1 W_2 \dots W_M} = \left\{ \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \right\} \tag{5.1.20}$$

Due to (5.1.11), let its *joint probability mass function* be

$$P_{W_1 W_2 \dots W_M} = \begin{cases} \dfrac{1}{\binom{M}{\mathcal{W}}} = \dfrac{(M-\mathcal{W})!W!}{\mathcal{W}!} & \mathbf{w}^k \in \Omega_{W_1 W_2 \dots W_M} \\ 0 & \text{Otherwise} \end{cases} \tag{5.1.21}$$

From (2.2.1), the *marginal probability mass function* of $W_1, W_2, \dots, W_M$ with sample space $\mathcal{D}_{W_i} = \{1, 0\}$ for $i = 1, ..., M$ is given by:

$$P_{W_i}(w) = \begin{cases} \dfrac{\binom{M-1}{\mathcal{W}}}{\binom{M}{\mathcal{W}}} = \dfrac{M-\mathcal{W}}{M} & w = 0 \\ \dfrac{\binom{M-1}{\mathcal{W}-1}}{\binom{M}{\mathcal{W}}} = \dfrac{\binom{M-1}{\mathcal{W}-1}}{\frac{M}{\mathcal{W}}\binom{M-1}{\mathcal{W}-1}} = \dfrac{\mathcal{W}}{M} & w = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{5.1.22}$$

## 5.2 Scores Random Vectors, $\mathbf{W}$

### 5.2.1 $\mathrm{E}[\mathbf{W}]$. Expectation vector

**$\mu_W$, expectation of $W_i$**

$$\mathrm{E}[W] = \sum_{i=1}^{2} w_i \, P(w_i) = 1 \, \frac{\mathcal{W}}{M} = \frac{\mathcal{W}}{M} = \mu_W \tag{5.2.1}$$

**$\mathrm{E}[\mathbf{W}]$, expected value of $\mathbf{W}$,**

Therefore, the expected value of a random vector $\mathbf{W}$ is:

$$\mathrm{E}[\mathbf{W}] = \begin{bmatrix} E[W_1] \\ E[W_2] \\ \vdots \\ E[W_M] \end{bmatrix} = \begin{bmatrix} \frac{\mathcal{W}}{M} \\ \vdots \\ \frac{\mathcal{W}}{M} \end{bmatrix} \tag{5.2.2}$$

### 5.2.2 $\mathbf{K}^F_{WW}$. Variance-Covariance Matrix

The matrix $\mathbf{K}^H_{WW}$ is defined in (2.3.8) then we will find the variance Var$W$ and the covariance Cov$(W_i, W_j) = \vartheta$.

**Var$W$ , variance of $W_i$**

$$\text{Var}(W) = \text{E}[W^2] - \text{E}[W]^2 \tag{5.2.3}$$

$$= \sum w_i^2 P(w_i) - \left(\frac{\mathcal{W}}{M}\right)^2 \tag{5.2.4}$$

$$= \frac{\mathcal{W}}{M} - \left(\frac{\mathcal{W}}{M}\right)^2 = \mu_W(1-\mu_W) \tag{5.2.5}$$

Thus the variance of the random variable $W$ is given by,

$$\text{Var}(W) = \mu_W(1-\mu_W) \tag{5.2.6}$$

**$\vartheta$, covariance of $W_i, W_j$**

From equations(5.1.21) and (5.1.22), the score random variable $W_i$ of NofM coding are not independent because:

$$\prod_{i=1}^{M} P_{W_i}(w) = \left(\frac{M-\mathcal{W}}{M}\right)^{M-\mathcal{W}} \left(\frac{\mathcal{W}}{M}\right)^{\mathcal{W}} \neq \frac{(M-\mathcal{W})!\mathcal{W}!}{M!} = P_{W_1 W_2 \ldots W_M} \tag{5.2.7}$$

To find the covariance (2.3.7), we must find the bivariate joint probability distribution $f(W_i = w_i, W_j = w_j)$, for $i, j = 1, 2, \ldots, M$ .

We have the following facts:

- Each random variable can take on $2$ same binary values: $0, 1$.
- For any combination of $w_i, w_j$ with $i, j = 1, \ldots M$ the joint probability distribution $f$ are the same.
- The support $\Omega_F$ which is the sample space consists of $|\Omega_F| = \binom{M}{\mathcal{W}}$ random possible pair outcomes that are equally likely, that is, the probability for each random outcome is $\frac{1}{|\Omega_F|}$.
- $f(W_i, W_j) = f(W_j, W_i)$

To establish the joint probability distribution $f$, we need a general method to find the frequency for the pairs outcomes. As we have the fact that the random pair outcomes are equally likely, we only need to count the number of times each pair repeats.

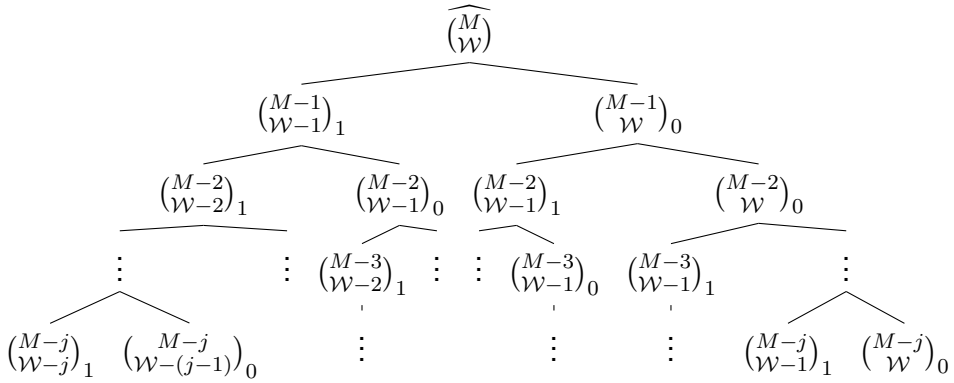
Let us introduce a notation to define the support $\Omega_F$ (equation (5.1.20)) as a matrix and from here we get the frequency hence the joint probability distribution.

**Definition 5.2.1.** Let $\widehat{\binom{M}{\mathcal{W}}}$ being a $M \times \binom{M}{\mathcal{W}}$ matrix. $\binom{M}{k}_1$ denote the number of $1$s and $\binom{M}{k}_0$ the number of $0$s at the matrix $\widehat{\binom{M}{\mathcal{W}}}$. $[\binom{M}{k}]$ denote a $M \times \binom{M}{k}$ matrix that represents all the combinations $M$ choose $k$. For $k = 0$ and $k = M$ we have that $[\binom{M}{0}]$ is a $M \times 1$ column vector of zeros, and $[\binom{M}{M}]$ is a $M \times 1$ column vector of ones.

To build the support $\Omega_F$ as a matrix, we first set the number one in the first row of the matrix and then set the zero. We know that the sum of ones and zeros is $|\Omega_F| = \binom{M}{\mathcal{W}}$. To know how many numbers one $(1's)$ we have, we use the combination formula for one less "1" and one less "element" of the vector. Using the notation above we will have $\binom{M-1}{\mathcal{W}-1}_1$ ones $(1's)$.

The number of zeros $(0's)$ is $\binom{M-1}{\mathcal{W}}_0$ since the size of the vector decreases but the ones $(1's)$ that are $W$, remain fixed. So we have $\binom{M-1}{\mathcal{W}-1}_1 + \binom{M-1}{\mathcal{W}}_0 = \binom{M}{\mathcal{W}}$. In the second row we set the ones and zeros again in the same way but for each row of ones and zeros, and so on.

In general, the matrix $\widehat{\binom{M}{\mathcal{W}}}$ can represent as a tree, where each line $j$ represents the $j$th row of the matrix




*Tree for NofM coding* (5.2.8)

Note that last row using the tree is when $\mathcal{W} = j$ but it is not the last row of the matrix. We define the matrices $[\binom{M}{k}]$, to denote the entire matrix in the definition (5.2.1).

For instance, with $M = 4, \mathcal{W} = 2$ we have

$$\Omega_{W_1W_2W_3W_4} = \widehat{\binom{4}{2}} = \begin{bmatrix} & \binom{3}{1}_1 & & \binom{3}{2}_0 \\ \binom{2}{0}_1 & & \binom{2}{1}_0 \ \binom{2}{1}_1 & & \binom{2}{2}_0 \\ [\binom{2}{0}] & & [\binom{2}{1}] \ [\binom{2}{1}] & & [\binom{2}{2}] \end{bmatrix} \tag{5.2.9}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{4\times\binom{4}{2}} \tag{5.2.10}$$

the first row corresponds to the first random variable $W_1$, the second row to the $W_2$ random variable and so on.

To find the frequency we can consider two first random variables $W_1, W_2$ in $\Omega_F$,

$$\Omega_{W_1 \dots W_M} = \widehat{\binom{M}{\mathcal{W}}} = \begin{bmatrix} \binom{M-1}{\mathcal{W}-1}_1 & \binom{M-1}{\mathcal{W}}_0 \\ \binom{M-2}{\mathcal{W}-2}_1 & \binom{M-2}{\mathcal{W}-1}_0 \quad \binom{M-2}{\mathcal{W}-1}_1 & \binom{M-2}{\mathcal{W}}_0 \\ \vdots & \vdots \quad \vdots & \vdots \end{bmatrix} \tag{5.2.11}$$

$$= \begin{bmatrix} \binom{M-1}{\mathcal{W}-1}_1 & \binom{M-1}{\mathcal{W}}_0 \\ f_3 & f_2 \quad f_2 & f_1 \\ \vdots & \vdots \quad \vdots & \vdots \end{bmatrix} \tag{5.2.12}$$

the frequency of the pair outcome $(1,1)$ is $\binom{M-2}{\mathcal{W}-2}_1 = f_3$, of the pair $(1,0) = (0,1)$ is $\binom{M-2}{\mathcal{W}-1}_0 = \binom{M-2}{\mathcal{W}-1}_1 = f_2$ and $(0,0)$ is $\binom{M-2}{\mathcal{W}}_0 = f_1$ therefore we have that,

$$f(W_1, W_1) = P_1 = \frac{f_1}{\binom{M}{\mathcal{W}}} = \frac{\binom{M-2}{\mathcal{W}}}{\binom{M}{\mathcal{W}}} \tag{5.2.13}$$

$$f(W_1, W_2) = P_2 = \frac{f_2}{\binom{M}{\mathcal{W}}} = \frac{\binom{M-2}{\mathcal{W}-1}}{\binom{M}{\mathcal{W}}} \tag{5.2.14}$$

$$f(W_2, W_2) = P_3 = \frac{f_3}{\binom{M}{\mathcal{W}}} = \frac{\binom{M-2}{\mathcal{W}-2}}{\binom{M}{\mathcal{W}}} \tag{5.2.15}$$

We sumarize the bivariate joint probability distribution $f$ in the following table,

| $f(W_i, W_j)$ | $0$ | $1$ | $f_{W_j}(W_j)$ |
|---|---|---|---|
| $0$ | $P_1$ | $P_2$ | $P_1 + P_2$ |
| $1$ | $P_2$ | $P_3$ | $P_2 + P_3$ |
| $f_{W_i}(W_i)$ | $P_1 + P_2$ | $P_2 + P_3$ | $1$ |

(5.2.16)

Now we can calculate the covariance(2.3.7). Doing the operations by rows, we have that Cov$(W_i, W_j) = c_0 + c_1$ where

$$c_0 = P_1\ (0 - \mu_W)(0 - \mu_W) + P_2\ (0 - \mu_W)(1 - \mu_W) \tag{5.2.17}$$

$$c_1 = P_2\ (1 - \mu_W)(0 - \mu_W) + P_3\ (1 - \mu_W)(1 - \mu_W) \tag{5.2.18}$$

Using the combination formula,

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

the following properties of the combination

$$\binom{n}{1} = n, \quad \binom{n}{0} = \binom{n}{n} = 1, \quad \binom{n}{n-r} = \binom{n}{r}, \quad \binom{n}{r} + \binom{n}{r-1} = \binom{n+1}{r} \tag{5.2.19}$$

and simplifying, we get,

$$\text{Cov}(W_i, W_j) = c_0 + c_1 \tag{5.2.20}$$
$$= P_1\ \mu_W^2 + P_2\ \mu_W(1-\mu_W) - P_2\ \mu_W(1-\mu_W) + P_3\ (1-\mu_W)^2 \tag{5.2.21}$$
$$= \mu_W^2 - 2\mu_W\ P_3 \frac{\mathcal{W}-1}{M-1} + P_3 \tag{5.2.22}$$
$$= \mu_W \frac{\mathcal{W}-1}{M-1} - \mu_W^2 \tag{5.2.23}$$

Thus the covariance $\text{Cov}(W_i, W_j)$ of scores of NofM is given by:

$$\text{Cov}(W_i, W_j) = \mu_W \frac{\mathcal{W}-1}{M-1} - \mu_W^2 \tag{5.2.24}$$

where $\mu_W$ is defined in the equation (5.2.1).

## 5.3 Integration Random Variable, $S$

### 5.3.1 Modulation vector function, $\Psi_F$

For (2.4.1), we denote $\Psi_F : \Omega \to \Xi_F$, the function for generating modulations vector for NoM coding. The function we chose in the present article is parameterized by the number $\mathcal{N} \in \{1, \ldots, M\}$ and defined as:

$$\Psi_F(\hat{\mathbf{w}}^k; \mathcal{N}) \equiv \Phi_F(\hat{\mathbf{w}}^k; \mathcal{N}) \tag{5.3.1}$$

Below, we consider $\mathbf{v}_F^1 = \Psi_F(\hat{\mathbf{w}}^1; \mathcal{N}) = (1, \ldots, 1_{\mathcal{N}}, 0, \ldots 0)^\mathsf{T}$, which contains $\mathcal{N}$ ones.

For NofM, the modulation vector gated up to the first $I$ components is then defined by

$$\mathbf{v}_{F,I}^1 = (1, \ldots, 1, 0, \ldots 0)^\mathsf{T} \tag{5.3.2}$$

which contains $I < \mathcal{N}$ ones.

### 5.3.2 $S_{F,\mathcal{N}}$. Integration-Final Potential $I = N$

$\mathrm{E}[S_{F,\mathcal{N}}]$, **expectation of final potential**

$$\mathrm{E}[S_{F,\mathcal{N}}] = \mathrm{E}[\mathbf{v}_{F,I}^\mathsf{T}\ \mathbf{W}] = \mathbf{v}_{F,I}^\mathsf{T}\, \mathrm{E}[\mathbf{W}] \tag{5.3.3}$$
$$= \mathbf{v}_{F,I}^\mathsf{T} \cdot \left(\frac{\mathcal{W}}{M}, ..., \frac{\mathcal{W}}{M}\right)^\mathsf{T} \tag{5.3.4}$$
$$= (1\ 1\ \ldots\ 0) \cdot \left(\frac{\mathcal{W}}{M}, ..., \frac{\mathcal{W}}{M}\right)^\mathsf{T}$$
$$= \left(\frac{\mathcal{W}}{M} + ... + \frac{\mathcal{W}}{M}\right) = \frac{\mathcal{N}\,\mathcal{W}}{M} \tag{5.3.5}$$

Then the expectation of Integration $S_{F,\mathcal{N}}$ is given by

$$\mathrm{E}[S_{F,\mathcal{N}}] = \frac{\mathcal{N}\,\mathcal{W}}{M} \tag{5.3.6}$$

$\mathrm{Var}[S_{F,\mathcal{N}}]$, **variance of final potential**

From (2.4.15), we have $\mathrm{Var}[S_{F,\mathcal{N}}] = \mathbf{v}_{F,\mathcal{N}}^{\mathsf{T}} \cdot \mathbf{K}_{WW}^{F} \cdot \mathbf{v}_{F,\mathcal{N}}$. The Variance-covariance matrix, $\mathbf{K}_{WW}^{F}$, is defined in equation (2.3.8), and the modulation vector $\mathbf{v}_{F,\mathcal{N}}^{1}$ of final potential for NofM coding is given by (5.3.2). Therefore,

$$\mathrm{Var}[S_{F,\mathcal{N}}] = \mathbf{v}_{F,\mathcal{N}}^{\mathsf{T}} \cdot \mathbf{K}_{WW}^{F} \cdot \mathbf{v}_{F,\mathcal{N}} \tag{5.3.7}$$

$$= (1\ 1 \ldots 1\ 0\ \ldots 0) \begin{bmatrix} \mathrm{Var}W & \vartheta & \ldots & \vartheta \\ \vartheta & \mathrm{Var}W & \ldots & \vartheta \\ \vdots & \vdots & \ddots & \vdots \\ \vartheta & \vartheta & \ldots & \mathrm{Var}W \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.3.8}$$

$$= \mathrm{Var}W + (\mathcal{N}-1)\,\vartheta + \ldots + \mathrm{Var}W + (\mathcal{N}-1)\,\vartheta \tag{5.3.9}$$

$$= \mathcal{N}\ \mathrm{Var}W + \mathcal{N}\ (\mathcal{N}-1)\vartheta \tag{5.3.10}$$

Then the variance of integration is given by:

$$\mathrm{Var}[S_{F,\mathcal{N}}] = \mathcal{N}\ \mathrm{Var}W + \mathcal{N}\ (\mathcal{N}-1)\ \mathrm{Cov}(W_i, W_j) \tag{5.3.11}$$

where $\mathrm{Var}W$ is given by (5.2.6) and $\mathrm{Cov}(W_i, W_j)$ by (5.2.24).

### 5.3.3 $S_{F,I}$. Integration-intermediate states, $I < \mathcal{N}$

$\mathrm{E}[S_{F,I}]$, **expectation at intermediate states**

$$\mathrm{E}[S_{F,I}] = \mathrm{E}[\mathbf{v}_{F,I}^{\mathsf{T}}\ \mathbf{W}] = \mathbf{v}_{F,I}^{\mathsf{T}}\,\mathrm{E}[\mathbf{W}] \tag{5.3.12}$$

$$= \mathbf{v}_{F,I}^{\mathsf{T}} \cdot \left(\frac{\mathcal{W}}{M}, ..., \frac{\mathcal{W}}{M}\right)^{\mathsf{T}}$$

$$= (1\ 1\ \ldots\ 0) \cdot \left(\frac{\mathcal{W}}{M}, ..., \frac{\mathcal{W}}{M}\right)^{\mathsf{T}}$$

$$= \left(\frac{\mathcal{W}}{M} + ... + \frac{\mathcal{W}}{M}\right) = \frac{I\ \mathcal{W}}{M} \tag{5.3.13}$$

Then the expectation of Integration $S_{F,I}$ is given by

$$\mathrm{E}[S_{F,I}] = \frac{I\ \mathcal{W}}{M} \tag{5.3.14}$$

$\mathrm{Var}[S_{F,I}]$, **variance at intermediate states**

From (2.4.15), the variance at intermediate states is given by $\mathrm{Var}[S_{F,I}] = \mathbf{v}_{F,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW}^{F} \cdot \mathbf{v}_{F,I}$. From (2.3.8), we have the Variance-covariance matrix $\mathbf{K}_{WW}^{F}$ and the modulation vector $\mathbf{v}_{F,I}^{1}$ for NofM code is given by (5.3.2). Therefore,

$$\mathrm{Var}[S_{F,I}] = \mathbf{v}_{F,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW}^{F} \cdot \mathbf{v}_{F,I}$$

$$= (1 \ldots 1\ 0\ \ldots 0) \begin{bmatrix} \mathsf{Var}W & \vartheta & \ldots & \vartheta \\ \vartheta & \mathsf{Var}W & \ldots & \vartheta \\ \vdots & \vdots & \ddots & \vdots \\ \vartheta & \vartheta & \ldots & \mathsf{Var}W \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.3.15}$$

$$= \mathsf{Var}W + (I-1)\,\vartheta + \ldots + \mathsf{Var}W + (I-1)\,\vartheta \tag{5.3.16}$$

$$= I\ \mathsf{Var}W + I\ (I-1)\vartheta \tag{5.3.17}$$

Then the variance of integration is given by

$$\mathrm{Var}[S_{F,I}] = I\ \mathsf{Var}W + I\ (I-1)\ \mathsf{Cov}(W_i, W_j) \tag{5.3.18}$$

where Var$W$ is given by (5.2.6) and Cov$(W_i, W_j)$ by (5.2.24).

### 5.3.4 Maximum value of final Potential, max($S_{F,\mathcal{N}}$)

The maximum value of integration $S_{F,\mathcal{N}}$ is defined in equation (2.4.18). Thus, for NofM coding, we have that $\mathbf{w}^1 = (1, \ldots, 1_{\mathcal{W}}, 0, \ldots, 0)$ and modulation vector is given in the equation (5.3.2), therefore,

$$\max(S_{F,\mathcal{N}}) = 1 + 1 \ldots + 1 = \mathcal{N} \tag{5.3.19}$$

considering $\mathcal{W} > \mathcal{N}$. If $\mathcal{W} < \mathcal{N}$, $\max(S_{F,\mathcal{N}}) = \mathcal{W}$ .

# 6 Ranked-NoM Coding

"Ranked-NoM" (R-NoM) is a novel proposal which combines features from both ROC (chapter V) and NoM (chapter VII) coding schemes: only the first $N$ input spikes among $M$ afferents are propagated, but their order is readout by downstream neurons thanks to inhomogeneous weights and linear desensitization.

In Ranked-NoM TTFS coding scheme both mechanism shunting inhibition and inhibitory circuit are implemented by a function which decrease linearly, although other schemes could also be explored using a similar approach in the context of sparse distributed memory (SDM) (e.g. geometric series as in [10]).

In this chapter we apply the TTFS coding theory (chapter IV) for Ranked-NoM (R-NoM) to establish the expectation and the variance for the integration at final potential and intermediate states. There are two stages, the first to develop the probabilistic theory of synaptic weights defined as random vectors $\mathbf{W}$ which we call **Ranked-NoM scores**. In section 6.1 we define the support of the scores $\Omega_H$ to find the joint $P_{W_1 W_2 \ldots W_M}$ and the marginal $P_{W_i}(w)$ probability mass functions. We can start directly at the scores thanks to the construction done in Section $1$ of Chapter IV where the rank order is already implicit.

In the section 6.2 we establish the expectation vector $\mathrm{E}[\mathbf{W}]$ and the variance-covariance matrix $\mathbf{K}_{WW}^{H}$. Since the random variables $W_1, W_2, \ldots, W_M$ are not independent we find the covariance $\vartheta$ using a tree method to calculate the frequency to bivariate probability distribution $f(W_i, W_j)$.

The second stage is to develop the statistical theory for the random variable integration $S_{H,I}$ at final $I = \mathcal{N}$ and intermediate states $I < \mathcal{N}$ which consists in computing scores vectors with modulation vector $\mathbf{v}_H^l$ by the inner product. Finally we define a formula for the maximum integration value, $\max(S_{H,\mathcal{N}})$, which will use to compare the codes by discriminability power.

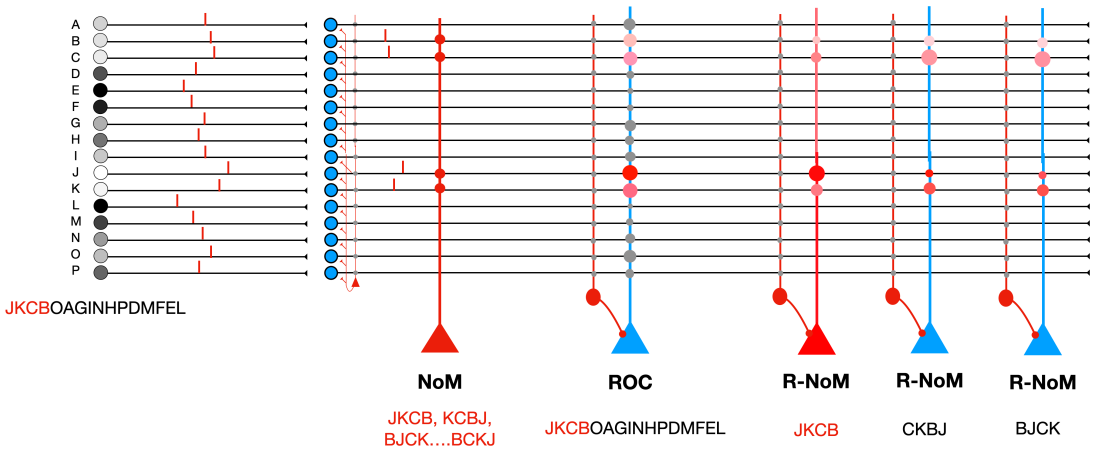



Figure 6.1 – Comparison of ROC, $N$-of-$M$ and Ranked-NoM codes. On the left, the $M = 16$ afferents fire in the order JKCBOAGINHPDMFEL, but a $4$-winner-take-all mechanism only lets the $N = 4$ first spikes through. NoM coding: the readout neuron uses binary weights: $W = 4$ ones, and $M - W = 12$ zeros. The final potential reaches the maximal value of $4$ if the $N$ first spikes correspond to the $W$ non-zero weights. The order of these $4$ first spikes does not matter. Rank Order coding **ROC.**: the neuron is set up to respond maximally to the order JKCBOAGINHPDMFEL, even though here only the $4$ input spikes are propagated. Ranked-NoM coding **R-NoM**:: we show three readout neurons that are selective to three different orders for the $4$ first spikes, among the $4! = 24$ possible orders, thanks to graded weights and modulations, both in $\{1, 2, 3, 4\}$.

## 6.1 Ranked-NoM scores-Support $\Omega_H$

### 6.1.1 Score vector function, $\Phi_H$

Ranked-NoM Coding is parametized by the number $\mathcal{W} \in \{1, \dots, M\}$. Under this coding, we want the best weights vector $\mathbf{w^1}$ made up of an arithmetic sequence, over the first $\mathcal{W}$ elements, starting from $\mathcal{W}$, with rate $-1$, down to zero, and the other elements set to zero.

The scores are obtained by the vector-value function $\Phi_H(\hat{\mathbf{w}}^k) = \mathbf{w}_H^l$ defined in (2.2.3). Then the vectorial components are given by,

$$\phi_{H,i}(\hat{\mathbf{w}}^k) = \phi_{H,i}(\hat{w}_1^k, \dots, \hat{w}_M^k) = \max(0, \hat{w}_i^k - M + \mathcal{W}) \tag{6.1.1}$$

such that:

$$\Phi_H(\hat{\mathbf{w}}^k) = \Phi_H(\hat{w}_1^k, \ldots, \hat{w}_M^k) \tag{6.1.2}$$
$$= \Phi_H(M - r_1^k, M - r_2^k, \ldots, M - r_M^k) \tag{6.1.3}$$
$$= (\phi_{H,1}(\hat{w}_1^k, \ldots, \hat{w}_M^k), \ldots, \phi_{H,M}(\hat{w}_1^k, \ldots, \hat{w}_M^k)) \tag{6.1.4}$$
$$= (\max(0, \mathcal{W} - r_1^k), \ldots, \max(0, \mathcal{W} - r_M^k))$$
$$= (w_1^l, \ldots, w_M^l) = \mathbf{w}_H^l \tag{6.1.5}$$

Note that for $k = 1$, we have $\hat{\mathbf{w}}^1 = (M, M-1 \ldots, 1)$ or equivalently $\mathbf{r}_i^1 = (0, \ldots, M-1)$

$$\Phi_H(\hat{\mathbf{w}}^1) = (\max(0, \mathcal{W}), \ldots, \max(0, \mathcal{W} - M + 1)) \tag{6.1.6}$$
$$= (\mathcal{W}, \mathcal{W} - 1, \ldots, 0) = \mathbf{w}_H^1 \tag{6.1.7}$$

which contains $M - \mathcal{W}$ zeros.

**Example 6.1.1.** As an illustration, let $M = 4$ and $\mathcal{W} = 2$ ($|\Omega| = 24$ permutations). We would have, for the best order:

$$\Phi_H(\hat{\mathbf{w}}^1) = (\max(0, 2), \max(0, 1), \max(0, 0), \max(0, -1)) \tag{6.1.8}$$
$$= (2, 1, 0, 0) = \mathbf{w}_H^1 \tag{6.1.9}$$

and for the worst order:

$$\Phi_H(\hat{\mathbf{w}}^{24}) = (\max(0, -1), \max(0, 0), \max(0, 1), \max(0, 2)) \tag{6.1.10}$$
$$= (0, 0, 1, 2) \tag{6.1.11}$$

Note that for $\hat{\mathbf{w}}^{18} = (2, 1, 3, 4)$, we would also obtain:

$$\Phi_H(\hat{\mathbf{w}}^{18}) = (\max(0, 0), \max(0, -1), \max(0, 1), \max(0, 2)) \tag{6.1.12}$$
$$= (0, 0, 1, 2) \tag{6.1.13}$$

therefore, $\Phi_H(\hat{\mathbf{w}}^{18}) = \Phi_H(\hat{\mathbf{w}}^{24})$.

Indeed, $\Phi_H$ maps different weights permutations onto the same score permutation. Hence, a subset of vectors that are pure internal permutation among negative or null values will map to the same element of $\Omega_H$. Since the cardinality of these subsets is the number of permutations of the $M - \mathcal{W}$ null elements, the cardinality of $\Omega_H$ is:

$$|\Omega_H| = \frac{M!}{(M - \mathcal{W})!} \tag{6.1.14}$$

In the illustrative example, $|\Omega_H| = \dfrac{4!}{(4-2)!} = 12$ permutations, and we would have $\Phi_H(\hat{\mathbf{w}}^{18}) = \Phi_H(\hat{\mathbf{w}}^{24}) = \mathbf{w}^{12}$.

### 6.1.2 Probability distributions functions

Let $\mathbf{W} = (W_1, W_2, \ldots, W_M)$ be the discrete random vector with support the ordered set $\Omega_H = \{\mathbf{w}^1, \ldots, \mathbf{w}^{\frac{M!}{(M-\mathcal{W})!}}\}$ which is generated by function $\hat{\Phi}_H$ defined in 6.1.1, and :

$$\Omega_H = \Omega_{W_1 W_2 \ldots W_M} = \left\{ \begin{bmatrix} \mathcal{W} \\ \mathcal{W}-1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} \mathcal{W}-1 \\ \mathcal{W} \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \ldots \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ \mathcal{W}-1 \\ \mathcal{W} \end{bmatrix} \right\} \tag{6.1.15}$$

Due to (6.1.14) let its *joint probability mass function* be

$$P_{W_1 W_2 \ldots W_M} = \begin{cases} \dfrac{1}{|\Omega_H|} = \dfrac{(M-\mathcal{W})!}{M!} & \mathbf{w}^k \in \Omega_H \\ 0 & \text{Otherwise} \end{cases} \tag{6.1.16}$$

and from (2.2.1), the *marginal probability mass function* of $W_1, W_2, \ldots, W_M$ with sample space $\mathcal{D}_{W_i} = \{\mathcal{W}, \mathcal{W}-1, ..., 1, 0\}$ for $i = 1, ..., M$, is given by:

$$P_{W_i}(w=0) = \frac{M-\mathcal{W}}{M} \tag{6.1.17}$$

$$P_{W_i}(w=k) = \frac{1}{M} \tag{6.1.18}$$

where $k = 1, ..., \mathcal{W}-1, \mathcal{W}$.

Therefore,

$$W_i = \begin{cases} \mathcal{D}_{W_i} = \{0, 1, ..., \mathcal{W}\} \\ \\ \mathrm{P}_{W_i}(w) = \begin{cases} \dfrac{M-\mathcal{W}}{M} & w = 0 \\ \dfrac{1}{M} & w = 1, 2, \ldots \mathcal{W} \end{cases} \end{cases} \tag{6.1.19}$$

## 6.2 Scores Random Vectors, $\mathbf{W}$

### 6.2.1 $\mathrm{E}[\mathbf{W}]$. Expectation vector

**$\mu_W$, expected value of $W_i$**

$$\mu_W \equiv \mathrm{E}[W_i] = \sum_{k=0}^{M} k\ P_{W_i}(k) = 0\frac{M-\mathcal{W}}{M} + \sum_{k=1}^{\mathcal{W}} k\ \frac{1}{M} \tag{6.2.1}$$

$$= \frac{1}{M}\sum_{k=1}^{\mathcal{W}} k$$

$$= \frac{1}{M}\frac{\mathcal{W}(\mathcal{W}+1)}{2} \tag{6.2.2}$$

**$\mathrm{E}[\mathbf{W}]$, expectation of $\mathbf{W}$**

Therefore,

$$\mathrm{E}[\mathbf{W}] = \begin{bmatrix} E[W_1] \\ E[W_2] \\ \vdots \\ E[W_M] \end{bmatrix} = \begin{bmatrix} \frac{\mathcal{W}(\mathcal{W}+1)}{2M} \\ \vdots \\ \frac{\mathcal{W}(\mathcal{W}+1)}{2M} \end{bmatrix} \tag{6.2.3}$$

### 6.2.2 $\mathbf{K}_{WW}^{H}$. Variance-Covariance matrix

The matrix $\mathbf{K}_{WW}^{H}$ is defined in (2.3.8) then we will find the variance Var$W$ and the covariance Cov$(W_i, W_j) = \vartheta$.

**Var$W$ , variance of $W_i$**

Following 6.1.19, we have:

$$\mathrm{E}[W_i^2] = \sum_{k=0}^{\mathcal{W}} k^2\ P_{W_i}(k) = 0^2\frac{M-\mathcal{W}}{M} + \sum_{k=1}^{\mathcal{W}} k^2\frac{1}{M} \tag{6.2.4}$$

$$= \frac{1}{M}\sum_{i=1}^{\mathcal{W}} k^2$$

$$= \frac{1}{M}\frac{\mathcal{W}(\mathcal{W}+1)(2\mathcal{W}+1)}{6} \tag{6.2.5}$$

Therefore,

$$\mathrm{Var}[W_i] = \mathrm{E}[W_i^2] - \mathrm{E}[W_i]^2 \tag{6.2.6}$$

$$= \frac{1}{M}\frac{\mathcal{W}(\mathcal{W}+1)(2\mathcal{W}+1)}{6} - \left(\frac{1}{M}\frac{\mathcal{W}(\mathcal{W}+1)}{2}\right)^2 \tag{6.2.7}$$

$$= \mu_W\ \frac{(2\mathcal{W}+1)}{3} - \mu_W^2 \tag{6.2.8}$$

### $\vartheta$, covariance of $W_i, W_j$

From equations(6.1.16) and (6.1.19), the score random variables $W_i$ of Ranked-NoM coding are not independent because:

$$\prod_{i=1}^{M} P_{W_i}(w) = \left(\frac{M-N}{M}\right)^{M-N}\left(\frac{1}{M}\right)^{N} \neq \frac{(M-N)!}{M!} = P_{W_1 W_2 \ldots W_M} \tag{6.2.9}$$

To find the covariance (2.3.7), we must find the bivariate joint probability distribution $f(W_i = w_i, W_j = w_j)$, for $i, j = 1, 2, \ldots, M$ .

We will use the following facts:

1. Each random variable $W_i$ can take on $\mathcal{W}$ same values: $0, 1, 2, \ldots, \mathcal{W}$.
2. The bivariate joint probability distribution $f(W_i, W_j)$ is the same for any $i, j = 1, \ldots, M$.
3. $f(W_i = k, W_j = k) = 0$ for $k \in \{1, 2, \ldots \mathcal{W}\}$
4. The support $\Omega_H$, which is the sample space, consists of $|\Omega_H| = \dfrac{M!}{(M-\mathcal{W})!}$ random possible pair outcomes that are equally likely, that is, the probability for each random outcome is $\dfrac{1}{|\Omega_H|}$.

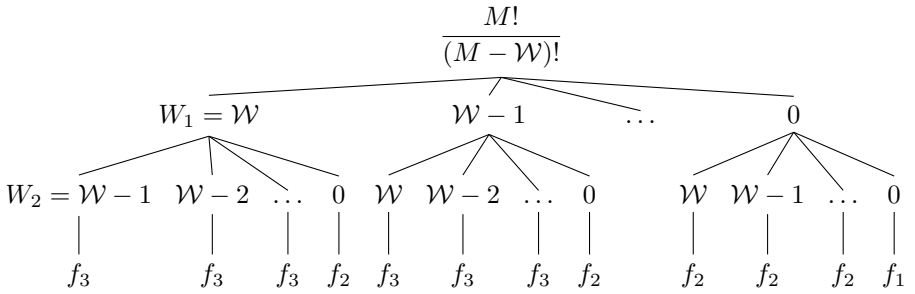
From fact (2), we only need to establish $f(W_1, W_2)$. From fact (3), we only need to establish the outcome frequency for each pair in:

$$(\mathcal{W}, \mathcal{W}-1), (\mathcal{W}, \mathcal{W}-2), \ldots, (\mathcal{W}-1, \mathcal{W}), (\mathcal{W}-1, \mathcal{W}-2), \ldots, (0, 0).$$

We first note that there is only three possibilities of outcome frequencies, that is, for the pairs: $(0,0), (0,y), (x,y)$ where $x, y \in \{\mathcal{W}, \mathcal{W}-1, \mathcal{W}-2, \ldots, 1\}$ are any integer different from zero. We denote :

$$(0,0)_f = f_1 \tag{6.2.10}$$
$$(0,y)_f = (y,0)_f = f_2 \tag{6.2.11}$$
$$(x,y)_f = (y,x)_f = f_3 \tag{6.2.12}$$

We can then enumerate the $|\Omega_H|$ possible outcomes of $\mathbf{W}$ within a tree, with first level corresponding to values taken by $W_1$, and second level corresponding to values taken by $W_2$ once $W_1$ is fixed, with leaves indicating the number of times each pair repeats:




*Tree Ranked-NoM code* (6.2.13)

For each branch, once values are given to $W_1$ and $W_2$, there is $(M-2)!$ permutations to consider, among which permutations corresponding to switching two null values should be considered the same. For a given branch, this number of permutations switching null values depend on the number of null already attributed to $W_1$ or $W_2$, hence:

$$f_1 = \frac{(M-2)!}{(M-\mathcal{W}-2)!} \tag{6.2.14}$$

$$f_2 = \frac{(M-2)!}{(M-\mathcal{W}-1)!} \tag{6.2.15}$$

$$f_3 = \frac{(M-2)!}{(M-\mathcal{W}-0)!} \tag{6.2.16}$$

Considering (from fact (4)) that the total number of possible pairs outcomes is $|\Omega_H| = \frac{M!}{(M-\mathcal{W})!}$, the corresponding probabilities are given by:

$$P_1 = \frac{f_1}{\frac{M!}{(M-\mathcal{W})!}} = \frac{\frac{(M-2)!}{(M-\mathcal{W}-2)!}}{\frac{M!}{(M-\mathcal{W})!}} = \frac{(M-\mathcal{W}-1)(M-\mathcal{W})}{(M-1)M} \tag{6.2.17}$$

$$P_2 = \frac{f_2}{\frac{M!}{(M-\mathcal{W})!}} = \frac{\frac{(M-2)!}{(M-\mathcal{W}-1)!}}{\frac{M!}{(M-\mathcal{W})!}} = \frac{M-\mathcal{W}}{(M-1)M} \tag{6.2.18}$$

$$P_3 = \frac{f_3}{\frac{M!}{(M-\mathcal{W})!}} = \frac{\frac{(M-2)!}{(M-W)!}}{\frac{M!}{(M-\mathcal{W})!}} = \frac{1}{(M-1)M} \tag{6.2.19}$$

We summarize the distribution $f$ in the following table:

| $f(W_i, W_j)$ | $0$ | $1$ | $2$ | $\dots$ | $\mathcal{W}$ | $f_{W_j}(w_j)$ |
|---|---|---|---|---|---|---|
| $0$ | $P_1$ | $P_2$ | $P_2$ | $\dots$ | $P_2$ | $P_1 + \mathcal{W}P_2$ |
| $1$ | $P_2$ | $0$ | $P_3$ | $\dots$ | $P_3$ | $P_2 + (\mathcal{W}-1)P_3$ |
| $2$ | $P_2$ | $P_3$ | $0$ | $\dots$ | $P_3$ | $P_2 + (\mathcal{W}-1)P_3$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\dots$ | $\vdots$ | $\vdots$ |
| $\mathcal{W}$ | $P_2$ | $P_3$ | $P_3$ | $\dots$ | $0$ | $P_2 + (\mathcal{W}-1)P_3$ |
| $f_{W_i}(w_i)$ | $P_1 + \mathcal{W}P_2$ | | | $\dots$ | $P_2 + (\mathcal{W}-1)P_3$ | $1$ |

(6.2.20)

Calculating the covariance defined in the equation (2.3.7) by rows $c_i$, from the bivariate joint probability table, we have that $\text{Cov}(W_i, W_j) = c_0 + c_1 + \cdots + c_{\mathcal{W}}$ where

$$
\begin{aligned}
c_0 &= P_1(0-\mu_W)(0-\mu_W) + P_2(0-\mu_W)(1-\mu_W) + \cdots + P_2(0-\mu_W)(\mathcal{W}-\mu_W) \\
c_1 &= P_2(1-\mu_W)(0-\mu_W) + P_3(1-\mu_W)(2-\mu_W) + \cdots + P_3(1-\mu_W)(\mathcal{W}-\mu_W) \\
c_{\mathcal{W}} &= P_2(\mathcal{W}-\mu_W)(0-\mu_W) + P_3(\mathcal{W}-\mu_W)(1-\mu_W) + \cdots + P_3(\mathcal{W}-\mu_W)(\mathcal{W}-1-\mu_W)
\end{aligned}
$$

Resolving the sum and simplifying, we get that the covariance for the scores random variable of Ranked-NoM coding is given by:

$$
\text{Cov}(W_i, W_j) = \frac{\mu_W}{M-1}\left(\mu_W - \frac{2\mathcal{W}+1}{3}\right) \tag{6.2.21}
$$

## 6.3 Integration Random Variable, $S$

### 6.3.1 Modulation vector function, $\Psi_H$

For (2.4.1), we denote $\Psi_H : \Omega \to \Xi_H$, the function for generating modulations vector for Ranked-NoM coding. The function we chose in the present article is parametized by the number $\mathcal{N} \in \{1, \ldots, M\}$ and defined as:

$$
\Psi_H(\hat{\mathbf{w}}^k; \mathcal{N}) \equiv \Phi_H(\hat{\mathbf{w}}^k; \mathcal{N}) \tag{6.3.1}
$$

Below, we consider $\mathbf{v}_H^1 = \Psi_H(\hat{\mathbf{w}}^1; \mathcal{N}) = (\mathcal{N}\ (\mathcal{N}-1) \ldots 1\ 0 \ldots 0)^{\mathsf{T}}$.

For Ranked-NoM coding, the modulation vector gated up to the first $I < \mathcal{N}$ components is then defined by

$$
\mathbf{v}_{H,I}^1 = (\mathcal{N}\ (\mathcal{N}-1) \ldots (\mathcal{N}-(I-1))\ 0 \ldots 0)^{\mathsf{T}} \tag{6.3.2}
$$

Then, for $I = \mathcal{N}$, that is, for the final potential, the modulation vector is given by

$$
\mathbf{v}_{H,\mathcal{N}}^1 = (\mathcal{N}\ (\mathcal{N}-1)\ \ldots\ 1\ 0\ \ldots\ 0)^{\mathsf{T}} \tag{6.3.3}
$$

For $I > \mathcal{N}$, we also take $\mathbf{v}_{H,\mathcal{N}}^1$.

### 6.3.2 $S_{H,\mathcal{N}}$. Integration-Final Potential $I = N$

$\mathrm{E}[S_{H,\mathcal{N}}]$, **expectation of final potential**

$$\mathrm{E}[S_{H,\mathcal{N}}] = \mathrm{E}[\mathbf{v}_{H,\mathcal{N}}^{\mathsf{T}}\,\mathbf{W}] = \mathbf{v}_{H,\mathcal{N}}^{\mathsf{T}}\,\mathrm{E}[\mathbf{W}] \tag{6.3.4}$$

$$= \mathbf{v}_{H,\mathcal{N}}^{\mathsf{T}} \cdot \frac{1}{M}\left(\frac{\mathcal{W}(\mathcal{W}+1)}{2}, ..., \frac{\mathcal{W}(\mathcal{W}+1)}{2}\right)^{\mathsf{T}} \tag{6.3.5}$$

$$= (\mathcal{N}\ (\mathcal{N}-1)\ \ldots\ 1\ 0\ \ldots\ 0) \cdot \frac{1}{M}\left(\frac{\mathcal{W}(\mathcal{W}+1)}{2}, ..., \frac{\mathcal{W}(\mathcal{W}+1)}{2}\right)^{\mathsf{T}} \tag{6.3.6}$$

$$= \frac{1}{M}\left(\frac{\mathcal{N}\mathcal{W}(\mathcal{W}+1)}{2} + \frac{\mathcal{W}(\mathcal{N}-1)(\mathcal{W}+1)}{2} + ... + \frac{\mathcal{W}(\mathcal{W}+1)}{2}\right)$$

$$= \frac{1}{M}\frac{\mathcal{W}(\mathcal{W}+1)}{2}(\mathcal{N} + (\mathcal{N}-1) + ... + 1)$$

$$= \frac{1}{M}\frac{\mathcal{W}(\mathcal{W}+1)}{2}\frac{\mathcal{N}(\mathcal{N}+1)}{2} \tag{6.3.7}$$

$$= \frac{1}{4M}\,\mathcal{W}(\mathcal{W}+1)\,\mathcal{N}(\mathcal{N}+1) \tag{6.3.8}$$

Then the expectation of Integration $\mathrm{E}[S_{H,\mathcal{N}}]$ is given by

$$\mathrm{E}[S_{H,\mathcal{N}}] = \frac{1}{4M}\,\mathcal{W}(\mathcal{W}+1)\,\mathcal{N}(\mathcal{N}+1) \tag{6.3.9}$$

$\mathrm{Var}[S_{H,\mathcal{N}}]$, **variance of final potential**

From (2.4.15), the variance of integration is given by $\mathrm{Var}[S_{H,\mathcal{N}}] = \mathbf{v}_{H,\mathcal{N}}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{H,\mathcal{N}}$. Using (2.3.8) for the expression of Variance-covariance matrix $\mathbf{K}_{WW}$ and the modulation vector $\mathbf{v}_{H,\mathcal{N}}^{1}$ for final potential given by (6.3.3), we have:

$$
\begin{aligned}
&\mathrm{Var}[S_{H,\mathcal{N}}] \\
&= \mathbf{v}_{H,\mathcal{N}}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{H,\mathcal{N}} && (6.3.10)\\
&= (\mathcal{N}\ (\mathcal{N}-1)\ \ldots\ 1\ 0\ \ldots\ 0)
\begin{bmatrix} \mathsf{Var}W & \vartheta & \ldots & \vartheta \\ \vartheta & \mathsf{Var}W & \ldots & \vartheta \\ \vdots & \vdots & \ddots & \vdots \\ \vartheta & \vartheta & \ldots & \mathsf{Var}W \end{bmatrix}
\begin{bmatrix} \mathcal{N} \\ \mathcal{N}-1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} && (6.3.11)\\
&= \mathcal{N}^2 \mathsf{Var}W + \mathcal{N}\ \vartheta \left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - \mathcal{N}\right) + (\mathcal{N}-1)^2 \mathsf{Var}W + && (6.3.12)\\
&(\mathcal{N}-1)\ \vartheta \left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - (\mathcal{N}-1)\right) + \cdots + \vartheta \left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - 1)\right) + \mathsf{Var}W \\
&= \mathsf{Var}W\ (\mathcal{N}^2 + (\mathcal{N}-1)^2 + \cdots + 1) + \vartheta\ \left[\mathcal{N}\left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - N\right)\right] + && (6.3.13)\\
&\vartheta\ \left[(\mathcal{N}-1)\left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - (\mathcal{N}-1)\right) + \cdots + \left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - 1\right)\right] \\
&= \mathsf{Var}W\ \left(\frac{\mathcal{N}(\mathcal{N}+1)(2\mathcal{N}+1)}{6}\right) + \vartheta\ \left[\sum_{i=1}^{\mathcal{N}} i \left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - i\right)\right] && (6.3.14)\\
&= \mathsf{Var}W\ \frac{\mathcal{N}(\mathcal{N}+1)}{2}\frac{2\mathcal{N}+1}{3} + \vartheta\ \left[\left(\frac{\mathcal{N}(\mathcal{N}+1)}{2}\right)^2 - \frac{\mathcal{N}(\mathcal{N}+1)}{2}\frac{2\mathcal{N}+1}{3}\right] && (6.3.15)
\end{aligned}
$$

Denoting $p = \dfrac{\mathcal{N}(\mathcal{N}+1)}{2}$ and $q = \dfrac{2\mathcal{N}+1}{3}$, it reads:

$$\mathrm{Var}[S_{H,\mathcal{N}}] = (p\ q)\mathsf{Var}W\ + (p^2 - p\ q)\ \mathsf{Cov}(W_i, W_j) \tag{6.3.16}$$

where $\mathsf{Var}W$ is given by (6.2.8) and $\mathsf{Cov}(W_i, W_j)$ by (6.2.21).

In case $\mathcal{N} = \mathcal{W}$, we have:

$$\mathrm{Var}[S_{H,\mathcal{N}}] = \frac{M^2\ \mu_W^2}{M-1}\ (q - \mu_W)^2 \tag{6.3.17}$$

where $\mu_W$ is given by (6.2.2).

### 6.3.3 $S_{H,I}$. Integration-intermediate states, $I < \mathcal{N}$

The modulation vector gated up to the first $I$ components is defined by $\mathbf{v}_{H,I}^1 = (\mathcal{N}\ (\mathcal{N}-1) \ldots (\mathcal{N}-(I-1))\ 0 \ldots 0)^{\mathsf{T}}$ in equation (6.3.2).

$\mathrm{E}[S_{H,I}]$, **expectation at intermediate states**

$$\begin{aligned}
\mathrm{E}[S_{H,I}] =& \mathrm{E}[\mathbf{v}_{H,I}^{\mathsf{T}}\ \mathbf{W}] = \mathbf{v}_{H,I}^{\mathsf{T}}\,\mathrm{E}[\mathbf{W}] && (6.3.18)\\
=& \mathbf{v}_{H,I}^{\mathsf{T}} \cdot \frac{1}{M}\left(\frac{\mathcal{W}(\mathcal{W}+1)}{2} \ldots \frac{\mathcal{W}(\mathcal{W}+1)}{2}\right)^{\mathsf{T}} && (6.3.19)\\
=& (\mathcal{N}\ (\mathcal{N}-1)\ldots(N-(I-1))\ 0\ldots 0)\cdot && (6.3.20)\\
& \frac{1}{M}\left(\frac{\mathcal{W}(\mathcal{W}+1)}{2} \ldots \frac{\mathcal{W}(\mathcal{W}+1)}{2}\right)^{\mathsf{T}}\\
=& \frac{\mathcal{W}(\mathcal{W}+1)}{2M}(\mathcal{N}+\mathcal{N}-1+\ldots+\mathcal{N}-(I-1)) && (6.3.21)\\
=& \frac{\mathcal{W}(\mathcal{W}+1)}{2M}\left(I\,\mathcal{N}-\frac{I(I-1)}{2}\right) && (6.3.22)
\end{aligned}$$

The expectation of Integration at intermediate states $S_{H,I}$ is then given by:

$$\mathrm{E}[S_{H,I}] = \frac{I\mathcal{W}(\mathcal{W}+1)}{4M}(2\,\mathcal{N}-I+1) \tag{6.3.23}$$

$\mathrm{Var}[S_{H,I}]$, **variance at intermediate states**

From (2.4.15), we have $\mathrm{Var}[S_{H,I}] = \mathbf{v}_{H,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{H,I}$. The Variance-covariance matrix $\mathbf{K}_{WW}$ is given by (2.3.8) and modulation vector $\mathbf{v}_{H,I}^{1}$ by (6.3.2). Therefore we have:

$$
\begin{aligned}
&\mathrm{Var}[S_{H,I}] \\
&= \mathbf{v}_{H,I}^{\mathsf{T}} \cdot \mathbf{K}_{WW} \cdot \mathbf{v}_{H,I} \\
&= (\mathcal{N}\ (\mathcal{N}-1) \ldots (\mathcal{N}-(I-1))\ 0 \ldots 0) \cdot \mathbf{K}_{WW} \cdot \begin{bmatrix} \mathcal{N} \\ (\mathcal{N}-1) \\ \vdots \\ (\mathcal{N}-(I-1)) \\ 0 \\ \vdots \\ 0 \end{bmatrix} && (6.3.24) \\
&= \mathsf{Var}W[\mathcal{N}^2 + (\mathcal{N}-1)^2 + \ldots + ((\mathcal{N}-(I-1))^2)] + \vartheta\,\mathcal{N} && (6.3.25) \\
&[\mathcal{N} + \mathcal{N} - 1 + \ldots + (\mathcal{N}-(I-1)] + \vartheta\ (\mathcal{N}-1) \\
&[\mathcal{N} + \mathcal{N} - 2 + \ldots + (\mathcal{N}-(I-1)] + \ldots + \vartheta\ (\mathcal{N}-(I-1)) \\
&[\mathcal{N} + \mathcal{N} - 1 + \ldots + (\mathcal{N}-(I-2))] \\
&= \mathsf{Var}W\left[\sum_{i=0}^{I-1}(\mathcal{N}-i)^2\right] + \vartheta\left[\sum_{i=0}^{I-1}(\mathcal{N}-i)(\mathcal{N}(I-1))\right] - && (6.3.26) \\
&\vartheta\left[\sum_{i=0}^{I-1}(\mathcal{N}-i)\left(\frac{I(I-1)}{2}-i\right)\right] \\
&= \mathsf{Var}W\left[\mathcal{N}^2 I - 2\,\mathcal{N}\frac{I(I-1)}{2} + \frac{I(I-1)}{2}\frac{(2\,I-1)}{3}\right] + \vartheta && (6.3.27) \\
&\left[\mathcal{N}\frac{I(I-1)}{2}(2\,\mathcal{N}-I+1)\right] - \vartheta \\
&\left[\frac{I(I-1)}{2}\left(\mathcal{N}(I-1) - \frac{I(I-1)}{2} + \frac{(2\,I-1)}{3}\right)\right]
\end{aligned}
$$

Defining $\hat{p} = \dfrac{I(I-1)}{2}$ and $\hat{q} = \dfrac{2\,I-1}{3}$ and simplifying, we obtain the formula for the variance at intermediate states:

$$
\begin{aligned}
\mathrm{Var}[S_{H,I}] = [\mathcal{N}(\mathcal{N}I - 2\hat{p}) + \hat{p}\,\hat{q}]\ \mathsf{Var}W + \\
\hat{p}\ [2\mathcal{N}(\mathcal{N}-I+1) + \hat{p} - \hat{q}]\ \mathsf{Cov}(W_i, W_j) && (6.3.28)
\end{aligned}
$$

**Maximum value of Final Potential, max($S_{H,\mathcal{N}}$)**

The maximum value max($S_{H,\mathcal{N}}$) of integration $S_{H,\mathcal{N}}$ is defined in equation (2.4.18). we have that $\mathbf{w}^1 = (\mathcal{W}, \mathcal{W}-1, \ldots, 1, 0, \ldots, 0)$ and modulation vector is given in the equation (**??**), therefore $\max(S_{H,\mathcal{N}})$ is,

$$\begin{aligned}
&=< \mathbf{w}^1, \mathbf{v}^1_{H,\mathcal{N}} > \\
&= \mathcal{W}\mathcal{N} + (\mathcal{W}-1)(\mathcal{N}-1) + \ldots + (\mathcal{W}-(\mathcal{N}-1))(\mathcal{N}-(\mathcal{N}-1)) \\
&= \mathcal{N}^2\mathcal{W} - (\mathcal{W}+\mathcal{N})\left(\frac{\mathcal{N}(\mathcal{N}+1)}{2} - \mathcal{N}\right) + \left(\frac{\mathcal{N}(\mathcal{N}+1)(2\mathcal{N}+1)}{6} - \mathcal{N}^2\right)
\end{aligned} \tag{6.3.29}$$

simplyfing, the expresion the max value of integration is given by,

$$\max(S_{H,\mathcal{N}}) = \mathcal{W}\mathcal{N}\left(\frac{\mathcal{N}+1}{2}\right) + \frac{\mathcal{N}(1-\mathcal{N}^2)}{6} \tag{6.3.30}$$

# 7 TTFS Coding Schemes Comparison

In the previous chapters we have established mathematical objects such as the measure of central tendency, $\mathrm{E}[S_{C,I}]$ and measure of dispersion $\mathrm{Var}[S_{C,I}]$ which describe the data set integration $S_{C,I}$ for each scheme. Therefore, in this chapter we define formulas to measure and compare the codes using these results which are summarised in the first section. We compare the coding schemes by their power to discriminate among stimuli by analizing the behavior for final potential and during propagation. We also explore speed-accuracy trade-off through numerical simulations, and we analyse what it implies to define the inputs of a set of permutations by the correlation coefficient of Pearson in the last section.

## 7.1 Synopsis of mathematical expressions

Let us summarise the results of the previous chapters in tables taking into account that the expectation $\mathrm{E}[S_{C,I}]$ and variance $\mathrm{Var}[S_{C,I}]$ of integration at intermediate states of each scheme $C$ depend on the mean $\mu_{W_C}$, variance Var$W_C$ and covariance Cov$_C(W_i, W_j)$ of the scores for the corresponding coding scheme $C$ Their full expressions are given in Table 7.1.

| $C$ | $\mu_{W_C}$ | Var$W_C$ | Cov$_C(W_i, W_j)$ |
|---|---|---|---|
| Ranked-NoM(H) | $\frac{\mathcal{W}(\mathcal{W}+1)}{2M}$ | $\mu_{W_H}\left(\frac{2\mathcal{W}+1}{3} - \mu_{W_H}\right)$ | $\frac{\mu_{W_H}}{M-1}\left(\mu_{W_H} - \frac{2\mathcal{W}+1}{3}\right)$ |
| NofM(F) | $\frac{\mathcal{W}}{M}$ | $\mu_{W_F}(1-\mu_{W_F})$ | $\mu_{W_F}\left(\frac{\mathcal{W}-1}{M-1} - \mu_{W_F}\right)$ |
| ROC(R) | $\frac{M+1}{2}$ | $\mu_{W_R}\left(\frac{M-1}{6}\right)$ | $\frac{\mu_{W_R}}{M-1}\left(\mu_{W_R} - \frac{2M+1}{3}\right)$ |

Table 7.1 – Formulas for the expectation, variance and covariance of the scores random variable $W$ for each scheme.

morever, for all schemes $C$, the expectation $\mathrm{E}[S_{C,I}]$ is a function of the mean $\mu_{W_C}$ and the variance $\mathrm{Var}[S_{C,I}]$ is a function of the variance Var$W_C$ and covariance Cov$_C(W_i, W_j)$ of scores of corresponding coding scheme $C$.

As a general pattern, we have the following linear combinations,

$$\mathrm{E}[S_{C,I}] = \lambda_C \, \mu_{W_C} \tag{7.1.1}$$

$$\mathrm{Var}[S_{C,I}] = \alpha_C \, \mathrm{Var}W_C + \beta_C \, \mathrm{Cov}_C(W_i, W_j) \tag{7.1.2}$$

the constants $\lambda_C, \alpha_C$ and $\beta_C$ depend on the parameters corresponding to its modulation vector $\mathbf{v}^1_{C,I}$. The following vectors are the $M \times 1$ modulation vector for each scheme $C$ at intermediate states $I$, and at the final potential $I = N$ for Ranked-NoM and NofM and $I = M$ for ROC,

| Ranked-NoM | | NofM | | ROC | |
|---|---|---|---|---|---|
| $0 < I < \mathcal{N}$ | $I = \mathcal{N}$ | $0 < I < \mathcal{N}$ | $I = \mathcal{N}$ | $0 < I < M$ | $I = M$ |
| $\begin{pmatrix} \mathcal{N} \\ \mathcal{N}-1 \\ \vdots \\ \mathcal{N}-(I-1) \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ | $\begin{pmatrix} \mathcal{N} \\ \mathcal{N}-1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1_I \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1_{\mathcal{N}} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ | $\begin{pmatrix} m^0 \\ m^1 \\ \vdots \\ m^{I-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ | $\begin{pmatrix} m^0 \\ m^1 \\ \vdots \\ m^{\mathcal{N}} \\ \vdots \\ m^{M-1} \end{pmatrix}$ |

The constants $\lambda_C, \alpha_C$ and $\beta_C$ for each scheme are provided in Table 7.2.

| $C$ | $\lambda_C$ | $\alpha_C$ | $\beta_C$ |
|---|---|---|---|
| Ranked-NoM(H) | $\frac{I(2\mathcal{N}-I+1)}{2}$ | $\mathcal{N}\,I(\mathcal{N}-I+1)+\frac{I(I-1)(2I-1)}{6}$ | $\mathcal{N}\,I(I-1)(\mathcal{N}-I+1)+\frac{I^2(I-1)^2}{4}-\frac{I(I-1)(2I-1)}{6}$ |
| NofM(F) | $I$ | $I$ | $I\,(I-1)$ |
| ROC(R) | $\frac{1-m^I}{1-m}$ | $\frac{1-m^{2I}}{1-m^2}$ | $\left(\frac{1-m^I}{1-m}\right)^2-\frac{1-m^{2I}}{1-m^2}$ |

Table 7.2 – Formulas for the expectation and variance coefficients of the different Integration schemes.

We also established the formula for the max value of activation for each scheme, thus the $\max(S_{C,I})$ for $\mathcal{W}>\mathcal{N}$, are given in Table 7.3.

| | $\max(S_{C,I})$ |
|---|---|
| Ranked-NoM(H) | $\mathcal{W}\mathcal{N}\left(\frac{\mathcal{N}+1}{2}\right)+\frac{\mathcal{N}(1-\mathcal{N}^2)}{6}$ |
| NofM(F) | $\mathcal{N}$ |
| ROC(R) | $\frac{(1-m)(1+M)-(1-m^{M+1})}{(1-m)^2}$ |

Table 7.3 – Formulas for the maximum value of integration $S_{C,I}$ for each scheme.

## 7.2 Discriminability measure

We define a measure of selectivity, which quantifies how much the neuron responds to its preferred pattern than to random stimulus. We define the preferred pattern by a *modulations vector* $\Psi_C(\mathbf{w}^1)$ which determine which input order is preferred by the efferent neurone under consideration.

Since $\mathbf{w}$ is a random vector, then $S_C(\mathbf{w},I)$ is a random function. Let $S_{C,I}$ denote the corresponding output random variable. Its distribution depends on the coding scheme. We compare the three coding schemes in terms of discriminative, characterizing its distribution by the difference between its best possible value and its expected values, scaled by its variance.

**Definition 7.2.1.** We define discriminability $D_C(I)$ as:

$$D_C(I)=\frac{\max(S_{C,I})-\mathrm{E}[S_{C,I}]}{\sqrt{\mathrm{Var}[S_{C,I}]}} \tag{7.2.1}$$

where $I \in \mathbb{Z}$ and takes values for ROC in the interval $[1, M]$ and for Ranked-NoM and NoM coding in the interval $[1, \mathcal{N}]$. Given that for values $\mathcal{N} < I < M$, Ranked-NoM and NoM are not defined, we set those values to the final integration corresponding to each scheme.

This discriminability is also known as the signal-to-noise ratio in other papers[27, 28, 19].

### 7.2.1 Behavior of discriminability for final potential

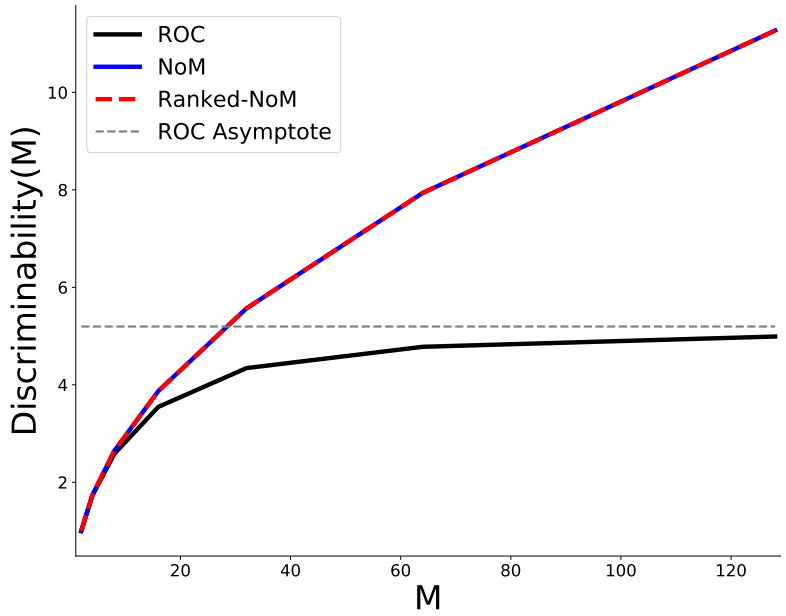



Figure 7.1 – Behavior of the maximal discriminability as a function of the total number of inputs $M$. NoM and Ranked-NoM (set to $\mathcal{W} = \mathcal{N} = M/2$) converge to the same maximal values and so the two curves overlap. For these two schemes, the maximal discriminability scales as $D_H(M) = D_F(M) = \sqrt{M-1}$. Hence, discriminability is not limited, and adding inputs will always improve it. By contrast, maximal discriminability for ROC saturates at an asymptotic value ($Y \simeq 5.2$ for $m = 0.8$ here).

Having established the complete expression of discriminability for the three schemes, we can now compare how they perform.

We first illustrate how the total number of available inputs ($M$) affects discriminability (Fig. 7.1).

To get the expressions of final potential for the three schemes we set $\mathcal{N} = \mathcal{W} = M/2$ for Ranked-NoM and NoM codes, for ROC we assign variables and simplify thus we get the expresions of final potential for each scheme:

**Rank Order coding**

We get the expression of final potential Let us assign $p = \frac{1-m^{2M}}{1-m^2}, q = \frac{1-m^M}{1-m}$ and substituting in the formulas,

$$\max(S_{R,M}) = \frac{(1-m)(1+M)-(1-m^{M+1})}{(1-m)^2} \tag{7.2.2}$$
$$= \frac{1+M}{1-m} - \frac{(1-m^{M+1})}{(1-m)^2} \tag{7.2.3}$$
$$= \frac{1+M}{1-m} - \frac{q+m^M}{1-m} \tag{7.2.4}$$
$$= \frac{M-q\,m}{1-m} \tag{7.2.5}$$

$$\mathrm{E}[S_{F,M}] = \lambda_F\ \mu_{W_F} \tag{7.2.6}$$
$$= \frac{1-m^M}{1-m}\ \frac{M+1}{2} \tag{7.2.7}$$
$$= q\ \frac{M+1}{2} \tag{7.2.8}$$

$$\mathrm{Var}[S_{R,\mathcal{N}}] = p\mathsf{Var}W\ + (q^2-p)\ \mathsf{Cov}(W_i, W_j) \tag{7.2.9}$$
$$= p\ \mu_{W_R}\ \left(\frac{M-1}{6}\right) + (q^2-p)\frac{\mu_{W_R}}{M-1}\ \left(\mu_{W_R} - \frac{2M+1}{3}\right) \tag{7.2.10}$$
$$= \frac{\mu}{6(M-1)}\left(p(M-1)^2 + 6(q^2-p)\left(\mu - \frac{2M+1}{3}\right)\right) \tag{7.2.11}$$
$$= \frac{M+1}{12}(p\ M - q^2) \tag{7.2.12}$$

then substituting in the formula of discriminability, max (7.2.5), expectation(7.2.8) and variance (7.2.12) of integration for ROC code, we get the following function,

$$D_F(M) = \frac{\max(S_{F,M}) - \mathrm{E}[S_{F,M}]}{\sqrt{\mathrm{Var}[S_{F,M}]}} \tag{7.2.13}$$
$$= \frac{\frac{M-q\,m}{1-m} - q\ \frac{M+1}{2}}{\sqrt{\frac{M+1}{12}(p\ M - q^2)}} \tag{7.2.14}$$
$$= \sqrt{\frac{\frac{(2(M-q\,m)-q(M+1)(1-m))^2}{4(1-m)^2}}{\frac{M+1}{12}(p\ M - q^2)}} \tag{7.2.15}$$
$$= \frac{\sqrt{3}}{1-m}\sqrt{\frac{(2(M-q\,m)-q(M+1)(1-m))^2}{(M+1)(p\ M - q^2)}} \tag{7.2.16}$$

dividing numerator and denominator of the radicand by $M^2$ to find the tendency of the function when $M \to \infty$, we get,

$$D_F(M) = \frac{\sqrt{3}}{1-m}\sqrt{\frac{\left(2\left(1-\frac{q\,m}{M}\right)-q\left(1+\frac{1}{M}\right)(1-m)\right)^2}{\left(1+\frac{1}{M}\right)\left(p-\frac{q^2}{M}\right)}} \tag{7.2.17}$$

Note that $\lim_{M\to\infty} p = \frac{1}{1-m^2}$ and $\lim_{M\to\infty} q = \frac{1}{1-m}$ then we have that,

$$\lim_{M\to\infty} D_F(M) = \lim_{M\to\infty} \frac{\sqrt{3}}{1-m}\sqrt{\frac{\left(2\left(1-\frac{q\,m}{M}\right)-q\left(1+\frac{1}{M}\right)(1-m)\right)^2}{\left(1+\frac{1}{M}\right)\left(p-\frac{q^2}{M}\right)}} \tag{7.2.18}$$

$$= \frac{\sqrt{3}}{1-m}\sqrt{\frac{(2-q(1-m))^2}{p}} \tag{7.2.19}$$

$$= \frac{\sqrt{3}}{1-m}\sqrt{1-m^2} \tag{7.2.20}$$

therefore for $m = 0.8$ the function $Y = D_F(M)$ has an asymptote horizontal in $Y = 5.2$ given that,

$$\lim_{M\to\infty} D_F(M) = \frac{\sqrt{3}}{1-0.8}\sqrt{1-0.8^2} = 5.196152422706633 \simeq 5.2 \tag{7.2.21}$$

see figure 7.1

### NofM

Substituting $\mathcal{N} = \mathcal{W} = M/2$, we have,

$$\max(S_{F,\mathcal{N}}) = \mathcal{N} = \frac{M}{2} \tag{7.2.22}$$

$$\mathrm{E}[S_{F,\mathcal{N}}] = \lambda_F\, \mu_{W_F} \tag{7.2.23}$$

$$= \mathcal{N}\frac{\mathcal{W}}{M} = \frac{M}{4} \tag{7.2.24}$$

thus for $\mathcal{N} = M/2$ and simplying we get, For the variance $\mathrm{Var}[S_{F,\mathcal{N}}]$, we have that

$$\mathrm{Var}W_F = \mu_{W_F}\,(1-\mu_{W_F}) = \frac{\mathcal{W}}{M}\left(1-\frac{\mathcal{W}}{M}\right) = 1/4$$

$$\mathrm{Cov}_H(W_i, W_j) = \frac{\mu_{W_H}}{M-1}\left(\mu_{W_H} - \frac{2\mathcal{W}+1}{3}\right) = \frac{1}{4(1-M)}$$

therefore,

$$\mathrm{Var}[S_{F,\mathcal{N}}] = \mathcal{N}\mathrm{Var}W \ + \mathcal{N}(\mathcal{N}-1)\ \mathrm{Cov}(W_i, W_j) \tag{7.2.25}$$

$$= \frac{M}{2}\ \frac{1}{4} + \frac{M}{4}(M-2)\ \frac{1}{4(1-M)} \tag{7.2.26}$$

$$= \frac{M^2}{16(M-1)} \tag{7.2.27}$$

then substituting in the formula of discriminability, max (7.2.22), expectation(7.2.24) and variance (7.2.27) of integration for NofM code, we get a function depending of $M$,

$$D_F(M) = \frac{\max(S_{F,\mathcal{N}}) - \mathrm{E}[S_{F,\mathcal{N}}]}{\sqrt{\mathrm{Var}[S_{F,\mathcal{N}}]}} \tag{7.2.28}$$

$$= \frac{\frac{M}{2} - \frac{M}{4}}{\sqrt{\frac{M^2}{16(M-1)}}} \tag{7.2.29}$$

$$= \sqrt{M-1} \tag{7.2.30}$$

see figure 7.1

**Ranked-NoM**

Let us define $p = \dfrac{\mathcal{N}(\mathcal{N}+1)}{2}$ and $q = \dfrac{2\mathcal{N}+1}{3}$, and substituting $\mathcal{N} = \mathcal{W}$, in each expression we get,

$$\max(S_{H,\mathcal{N}}) = \mathcal{W}\mathcal{N}\left(\frac{\mathcal{N}+1}{2}\right) + \frac{\mathcal{N}(1-\mathcal{N}^2)}{6} \tag{7.2.31}$$

$$= \mathcal{N}\mathcal{N}\left(\frac{\mathcal{N}+1}{2}\right) + \frac{\mathcal{N}(1+\mathcal{N})}{2}\frac{(1-\mathcal{N})}{3} \tag{7.2.32}$$

$$= \mathcal{N}\ p + p\ \frac{1-\mathcal{N}}{3} \tag{7.2.33}$$

$$= p\ \left(\mathcal{N} + \frac{1-\mathcal{N}}{3}\right) \tag{7.2.34}$$

$$= p\left(\frac{2\mathcal{N}+1}{3}\right) \tag{7.2.35}$$

$$= p\ q \tag{7.2.36}$$

$$\mathrm{E}[S_{H,\mathcal{N}}] = \lambda_C\ \mu_{W_C} \tag{7.2.37}$$

$$= \frac{\mathcal{N}(\mathcal{N}+1)}{2} \cdot \frac{\mathcal{W}(\mathcal{W}+1)}{2M} \tag{7.2.38}$$

$$= \frac{\mathcal{N}(\mathcal{N}+1)}{2} \cdot \frac{\mathcal{N}(\mathcal{N}+1)}{2M} \tag{7.2.39}$$

$$= \frac{p^2}{M} \tag{7.2.40}$$

For the variance $\mathrm{Var}[S_{H,\mathcal{N}}]$ we have that

$$\mathsf{Var}W_H = \mu_{W_H}\left(\frac{2\mathcal{W}+1}{3} - \mu_{W_H}\right) = \mu(q-\mu)$$

$$\mathsf{Cov}_H(W_i, W_j) = \frac{\mu_{W_H}}{M-1}\left(\mu_{W_H} - \frac{2\mathcal{W}+1}{3}\right) = \frac{\mu}{M-1}(\mu - q)$$

therefore,

$$\mathrm{Var}[S_{H,\mathcal{N}}] = (p\ q)\mathsf{Var}W\ + (p^2 - p\ q)\ \mathsf{Cov}(W_i, W_j) \tag{7.2.41}$$

$$= p\ q\ \mu(q-\mu) + (p^2 - p\ q)\ \frac{\mu}{M-1}(\mu - q) \tag{7.2.42}$$

$$= \frac{p\ \mu(q-\mu)(M\ q-p)}{M-1} \tag{7.2.43}$$

then substituting in the formula of discriminability, max (7.2.36), expectation(7.2.40) and variance (7.2.43) of integration for Ranked-NoM code, we get,

$$D_H(M) = \frac{\max(S_{H,\mathcal{N}}) - \mathrm{E}[S_{C,\mathcal{N}}]}{\sqrt{\mathrm{Var}[S_{C,\mathcal{N}}]}} \tag{7.2.44}$$

$$= \frac{p\ q - \dfrac{p^2}{M}}{\sqrt{\dfrac{p\ \mu(q-\mu)(M\ q-p)}{M-1}}} \tag{7.2.45}$$

$$= \sqrt{\frac{(p\ q - p^2/M)^2}{\dfrac{p\ \mu(q-\mu)(M\ q-p)}{M-1}}} \tag{7.2.46}$$

$$= \sqrt{\frac{p^2/M^2(qM-p)^2}{\dfrac{p\ \mu(q-\mu)(M\ q-p)}{M-1}}} \tag{7.2.47}$$

$$= \sqrt{\frac{(qM-p)(M-1)\ p}{M^2\mu(q-\mu)}} \tag{7.2.48}$$

$$D_H(M) = \sqrt{M-1} \tag{7.2.49}$$

see figure 7.1

We get the same function for NofM and Ranked-NoM:

$$D_H(M) = D_F(M) = \sqrt{M-1} \tag{7.2.50}$$

In light of these behaviors, we propose that Ranked-NoM and NoM are to be preferred over ROC.

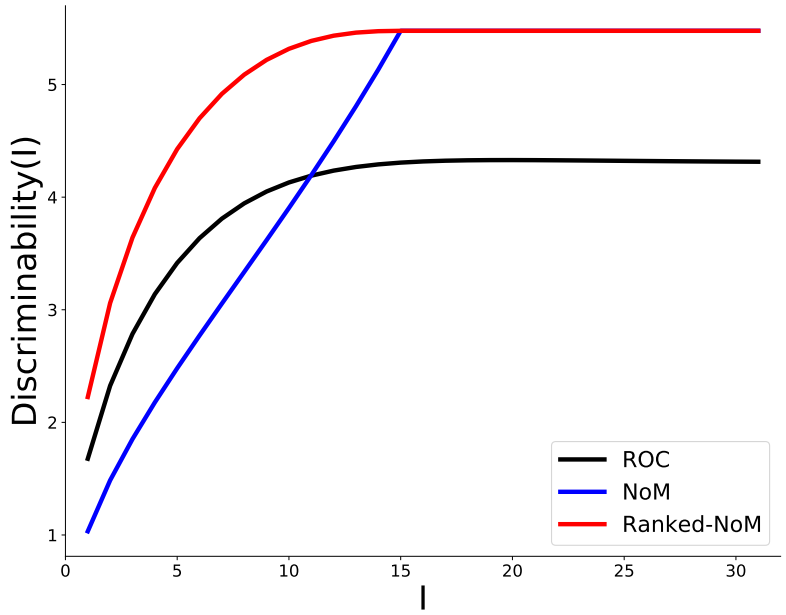



Figure 7.2 – Comparison of discriminability for the three coding schemes during propagation. The Discriminability, $D_C(I)$, (eq. 7.2.1) is reported as the number of inputs $I$ builds up, for each coding scheme: ROC (black), NoM (blue) and Ranked-NoM (red). For ROC coding, the inputs $I$ accumulate up to the maximal number (here, $M = 31$) while, in the two others, propagation stops beyond $\mathcal{N} = 15$ (in this case, we retain the value $D_C(\mathcal{N})$ for later values).

### 7.2.2 Behavior of discriminability during propagation

We now contrast, for a given $M = 31$, how discriminability increases as more and more inputs become available (namely, potential integration, Fig. 7.2).

As shown above, discriminability saturates to the same value for Ranked-NoM and NoM (here, $\mathcal{N} = \mathcal{W}$), while, for ROC, it saturates at a lower value, which depends on the ROC-parameter $m$ (here $m = 0.8$).

We also observe that NofM performs poorly early on since discriminability increases nearly linearly, while both ROC and Ranked-NoM increase more like an exponential relaxation to the final value.

In contrast to NofM, Ranked-NoM Coding then displays a much faster increase in discriminability in the early phase of input integration and reaches a higher value than ROC.

In this regard, Ranked-NoM displays the best performance, with a high discriminability for the very early inputs.

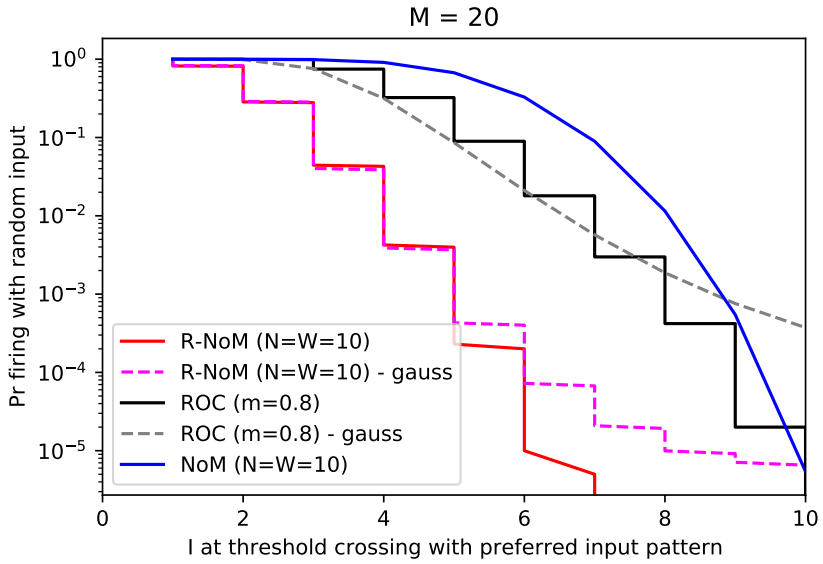



Figure 7.3 – Speed-accuracy trade-off. Here, we represented, for the different codes, the probability of firing to random inputs as a function of $I$ at threshold crossing for the preferred pattern (both variables depend on the threshold which is varied, but not represented on this figure). Solid lines indicate the probabilities estimated by Monte Carlo simulations (n=$2.10^5$ runs for each value). Dashed lines indicate the probability assuming a Gaussian distribution for the potential, with the mean and variance computed from Table 7.2 formulas. These Gauss-based values match well the simulations for low latencies. For higher latencies, they tend to overestimate the FA rate. This suggests that the potential is roughly normally distributed everywhere except in the right tail of the distribution.

## 7.3 Exploring the speed-accuracy trade-off through simulations

Importantly, our discriminability measure (Eq. 7.2.1) is based on the unconstrained membrane potential, i.e. ignoring the threshold. But of course, in a real scenario, a threshold is needed, especially for neurons in the hidden layers (otherwise, they will not fire!). When choosing a threshold, a high value:

- ensures that the probability of reaching it with random input (which may be seen as a false alarm, FA) is low.
- causes a longer latency even when the preferred pattern is given as input.

Conversely, a low threshold does the opposite (shorter latency but higher FA rate). This can be seen as a speed-accuracy trade-off.

We explored this trade-off through numerical simulations. We fixed $M = 20$ and estimated the false alarm probabilities for ROC ($m = 0.8$), R-NoM ($\mathcal{W} = \mathcal{N} = 10$) and NofM (also $\mathcal{W} = \mathcal{N} = 10$), as a function of the threshold, using $2.10^5$ random input spike patterns. In Figure 7.3, we plotted those probabilities as a function of the latency

(expressed in input spike number, not in seconds) for the firing response to the preferred pattern (latency which in turn depends on the threshold). This plot confirms the supremacy of R-NoM, especially in the early stage of the response, in agreement with Figure 7.2.

For example, here the preferred pattern has $\mathcal{N} = 10$ spikes. Let's say we want the receiver neuron to fire as soon as the fifth input spike is received. For R-NoM, this means the threshold should be in the $[294, 330]$ range. Choosing $330$ will minimize the FA rate, which will be around $3.10^{-4}$. For ROC, the corresponding threshold would be $28.36$, leading to a much higher FA rate of $0.1$. Finally, for NoM, the threshold would be $5$, and the FA rate $0.7$, which would be totally unacceptable.

Here again, our attempt to speculate upon how to combine computation-power of float-based TTFSs schemes and power-saving integer-based TTFS schemes offers a promising avenue: FA rate could be cut by a factor of three orders of magnitude compared with the former, and four orders compared with the latter.

## 7.4 Shannon Information content

we will consider an information theoretic view and characterize the Shannon information content contained in the different coding schemes. The information content of a Ranked-NoM code is

$$\log_2\left(\frac{M!}{(M-N)!}\right) \tag{7.4.1}$$

bits, which is $\log_2(N!)$ greater than the information content of a NofM code. We can find the information content $I$ (measured in bits) associated with a particular threshold $\delta$ by counting the number of codes $c(\delta)$ whose scalar product with the reference code is greater than or equal to the chosen threshold $\delta$. [**article**]

$$I = \log_2\left(\frac{M!}{(M-N)!}\right) - \log_2(c(\delta)) \tag{7.4.2}$$

the information content $I = \log_2\left(\frac{M!}{(M-N)!}\right)$ for $c(\delta) = 1$ Thus, the information content can be increased by increasing $M$ while holding constant $N$, so keeping the activity and, in a physical system, the power requirement constant.

## 7.5 Pearson correlation coefficient of the scores

The scores of the Ranked-NoM, NoM and ROC codes are negatively correlated and its pearson correlation coefficient $\rho$ are the same for the three schemes. They are not independent since permutation of a given set of values implies correlation. That is, the correlation comes from the fact that the choice for, say the last value to be put at the last position of the vector, depends upon the value which has not been put into the vector yet.

The correlation coefficient is define by:

$$\rho = \frac{\mathrm{Cov}(W_i, W_j)}{\sqrt{\mathrm{Var}W_i}\sqrt{\mathrm{Var}W_j}} \tag{7.5.1}$$

then we calculate $\rho$ for each scheme. Thus, for the Ranked-NoM code,

$$\rho_H = \frac{\frac{\mu_W}{M-1}(\mu_W - \frac{2\mathcal{W}+1}{3})}{\left(\sqrt{\mu_W \frac{2\mathcal{W}+1}{3} - \mu_W^2}\right)^2} \tag{7.5.2}$$

$$= \frac{\frac{1}{M-1}\left(\mu_W - \frac{2\mathcal{W}+1}{3}\right)}{\frac{2\mathcal{W}+1}{3} - \mu_W} = \frac{-1}{M-1} \tag{7.5.3}$$

for ROC code we have that,

$$\rho_R = \frac{\frac{\mu_W}{M-1}(\mu_W - \frac{2M+1}{3})}{\left(\sqrt{\frac{\mu_X}{6}(M-1)}\right)^2} \tag{7.5.4}$$

$$= \frac{\frac{1}{M-1}\left(\mu_W - \frac{2M+1}{3}\right)}{\frac{1}{6}(M-1)} \tag{7.5.5}$$

$$= \frac{2(3\mu_W - 2M - 1)}{(M-1)^2} = \frac{-1}{M-1} \tag{7.5.6}$$

and for NofM,

$$\rho_F = \frac{\mu_W\left(\frac{\mathcal{W}-1}{M-1}\right) - \mu_W^2}{(\sqrt{\mu_W(1-\mu_W)})^2} \tag{7.5.7}$$

$$= \frac{\left(\frac{\mathcal{W}-1}{M-1}\right) - \mu_W}{1-\mu_W} \tag{7.5.8}$$

$$= \frac{-1 + \frac{\mathcal{W}}{M}}{(M-1)\left(1 - \frac{\mathcal{W}}{M}\right)} = \frac{-1}{M-1} \tag{7.5.9}$$

therefore,

$$\rho_H = \rho_R = \rho_F = \frac{-1}{M-1} = \frac{1}{1-M} \tag{7.5.10}$$

Note that the correlation vanishes when $M->\infty$, and it is because the effect of previous choice tends to zero for next choice. So, it is the fact that we make permutations of a given set of values.

# 8 Conclusion and Perspectives

## 8.1 Summary of the contributions

### 8.1.1 Theoretical framework

In this research work, we presented a new mathematical framework which allows unifying various TTFS codes and propose a new code that we dubbed "Ranked-NoM" (R-NoM), which leverage two key features: sparsity and timing by integer-based property.

The probabilistic theory of the synaptic weights of TTFS codes as random vectors was developed where the input spikes' contribution depends on their arrival ranks rather than on their precise times.

This framework introduces the concept of modulation, in order to model the idea that the first input spikes contribute more, while later input spikes contribute less, or not at all. This was implemented by the dot product of the modulation vector with the synaptic weights which is the net contribution of each input spike to the neuron's potential: integration $S_{C,I}$. The modulation vector can also have a cut-off so that the last spikes make no contribution at all. This broad definition of modulation encompasses previous proposals (ROC, NoM), our new Ranked-NoM as well as new ones.

The integration is maximal when the spikes arrive in the order of the weights: the first spike should arrive through the strongest weight, and so on. This defines the preferred input spike pattern of a neuron. Then we defined discriminability, which measures how much more the neuron responds to its preferred pattern than to random inputs. Our framework allows us to compute this discriminability analytically. Thus, various TTFS codes can easily be compared in terms of discriminability. The framework also allows the design of new codes that maximize this discriminability.

Therefore, our main result is the theoretical framework we have developed in order to formalize how TTFS coding schemes behave, where the overall mathematical formulation for the first two moments of the distribution of potential, at any stage of the propagation, and for the three schemes under consideration has never been done before.

Theoretical neuroscience is based on the belief that methods of mathematics, physics, and computer science can provide important insights into nervous system function. Unfortunately, mathematics can sometimes seem more of an obstacle than an aid to understanding. Keeping this in mind, we built the theory by using the fundamental and basic concepts of multivariate calculus, multivariate probability theory and notions of combinatorial analysis, in a way that would keep development simple, clear, and sequential while employing the level of analysis necessary to be precise and rigorous. In this way it will serve to create new applications and will be the base to development in future studies using other TTFS schemes.

Theoretical progress opens new fields of applications, and in turn applications lead to new problems and fruitful research.

### 8.1.2 Results of comparison of schemes

We demonstrated that R-NoM has much more discriminative power than ROC and NoM, especially in the early phase of the response, which is already very selective. Thus it allows detectors that are both accurate and reactive. In addition, the fact that R-NoM uses only integers makes it much more hardware-friendly than ROC, and the geometric modulation suggested in [10]. We are pretty confident that R-NoM is totally underestimated so far: the fact that it outperforms ROC and NoM, both for the course of integration, and the speed-accuracy tradeoff, and by such a large extent, came as a surprise. Furthermore, we think that the fact that R-NoM has much more discriminative power than ROC and NoM, especially in the early phase of the response, is completely a new result.

We have validated and illustrate our theoretical results by Monte Carlo simulations which is a type of computational algorithm that uses repeated random sampling to obtain the likelihood of a range of results of occurring.

## 8.2 Perspectives

### 8.2.1 Learning Algorithm

One important issue that we did not address in this work is learning. We plan to address it in future research work. Only then we will be able to confront the different coding schemes with real-world data (e.g. CIFAR, ImageNet, Google Speech Commands) and compare their performance, possibly using the methodology of [**Guo2021**]. For unsupervised learning, we think that the STDP-like learning rule that we proposed in [**Thorpe2019**] could be adapted for the integer, non-binary, weights that are required for R-NoM. In short, part of the weights from unused synapses could be moved to used but not saturated synapses. For supervised learning, backpropagation has already been adapted to TTFS codes [32, 58, 21, 37, 46, 57, 5, 31]. Yet none of these approaches included the concept of a spike-based decreasing modulation. We will explore that possibility in future work.

### 8.2.2 Hardware implementations

There are however situations where NoM coding can be particularly interesting for hardware implementations. The advantages of Ranked-NoM coding described here apply in situations where incoming spikes are processed one by one. However, in some designs, it is possible to process spikes as a packet. For example, you could define an input array with M bits that are initially all set to zero. As spikes come in, the corresponding input lines can be flipped on until a fixed number of bits (N) are set to one. At this point, it is easy to determine the level of activation of a target neuron by performing a logical AND operation between the array of input spikes and a second array of bits corresponding to the connected weights (mathematically, it is the dot product between two vectors which we used in our theoretical framework). Counting the number of "hits" and comparing the result to the neuron's threshold can be done in a single clock cycle with specialized FPGA or ASIC hardware. Similar results can be obtained using memristor-based cross-bar arrays.

The current analysis provides a strong argument for using implementations that process incoming spikes in order since it is the only way to take advantage of the remarkable early discriminative power of Ranked-NoM coding. Such an approach goes a long way towards ensuring that computations can be done with the minimum number of spiking events.

To sum up, we have compared $3$ schemes. One is Rank-Order Coding (ROC), while powerful, ROC is hard to implement in hardware because it requires both precisely defined synaptic weights and an accurate desensitization mechanism. The two other schemes are much simpler to implement in hardware. In $N$-of-$M$ (NoM) coding, only the first $N$ neurons from $M$ channels fire. It allows highly selective responses despite only requiring binary synapses but does not take into account the precise order of firing. Our new proposal: Ranked-$N$-of-$M$ (R-NoM) coding, that combines features of both ROC and $N$-of-$M$ coding. Only $N$ neurons are allowed to fire, but the order in which they fire can be decoded by a hardware-friendly combination of integer weights and a progressive reduction in sensitivity that is strictly linear. It has a number of very interesting features including the ability to make very rapid decisions on the basis of very few spikes. The relevance for hardware design could potentially be very high.

### 8.2.3 Future research works

Confronting the different coding schemes with real world data is important (as [17] did, now cited on line $290$). This would require further research, first to design a learning algorithm. We also think that backprop and STDP could be adapted to the schemes we propose (including Ranked-NoM), and this will be a topic for future publications.

## 8.3 II part of theoretical framework

### 8.3.1 Skewness and kurtosis

The current mathematical framework specifies the first two moments of the potential distribution at intermediate states and final potential. These are the parameters which let us computed the discriminability measure. To compute speed-accuracy trade off (we only

explored it using numerical simulations), we need to know the probability distribution of the potential.

There are "methods" to obtain a probability distribution from its moments under some conditions. The moments of a random variable can be easily computed by using either its moment generating function, if it exists, or its characteristic function. We must be calculated the third statistical moment: Skewness, which measures how asymmetric the distribution is about its mean and the fourth statistical moment: kurtosis, which measures the amount in the tails and outliers.

Having their moments we need to prove if the conditions are satisfied: If the sequence of moments doesn't grow too quickly, then the distribution is determined by its moments. One sufficient condition is that if the moment generating function of a random variable has positive radius of convergence, then that random variable is determined by its moments. See Billingsley, Probability and Measure, chapter 30 [3].

### 8.3.2 Shannon information content

We can find the information content $I$ (measured in bits) associated with a particular threshold $\delta$ for each coding scheme by counting the number of codes $c(\delta)$ whose scalar product with the modulation vector is greater than or equal to the chosen threshold. We think that we can use the advantage to have the expectation and the variance of the distribution or in fact the distribution to get $c(\delta)$. Then we would have Information-speed-accuracy trade off.

### 8.3.3 Complete statistics learning theory

Complete statistics learning theory proposed by Vladimir Vapnik in 2020 [54], give us another way to build an algorithm for learning. Here I will show an example how we can use this theory of learning. It could be a way to implement in a power constrain way.

The following example is the classical problem of optimization:

**Example 8.3.1.** A rectangular box with a lid is to be made from $15$ m$^2$ of cardboard. Find the maximum volume of such a box. Thus, we have the following equations:

$$V(x,y,z) = xyz, \quad 2xy + 2yz + 2xz = 15 \tag{8.3.1}$$

and two optimization ways:

1. We optimize one function. We can express $V$ as a function of two variables,

$$V(x,y) = xy\left(\frac{15-2xy}{2y+2x}\right) \tag{8.3.2}$$

2. We optimize the volume function

$$V(x,y,z) = xyz \tag{8.3.3}$$

subject to the constraint:

$$g(x,y,z) = 2xy + 2yz + 2xz = 15 \tag{8.3.4}$$

which is the method of Lagrange Multipliers.

Roughly speaking, Vapnik propose the second way of optimization which he call "constrained optimization problem".

In a neural network we want to optimize the error(composition of functions) of our prediction for each output $Y_i$ and training sample $X^n$ as compared to the know label $t_j$

$$E^n = \frac{1}{2}\sum_j (f\left(\sum w_{ji} X_j^n\right) - t_j^n)^2 \tag{8.3.5}$$

The delta rule ($\delta_j^n = \frac{\partial E^n}{\partial a_j^n}$) is a gradient descent learning rule for updating the weights of the inputs to artificial neurons in a single-layer neural network

$$\frac{\partial E^n}{\partial w_{ij}} = \frac{\partial E^n}{\partial e_j^n}\frac{\partial e_j^n}{\partial Y_j^n}\frac{\partial Y_j^n}{\partial a_j^n}\frac{\partial a_j^n}{\partial w_{ij}} \tag{8.3.6}$$

$$\triangle w_{ji} = -\gamma \frac{\partial E^n}{\partial w_{ij}} \tag{8.3.7}$$

Therefore the constrained optimization problem is given by:

$$\begin{aligned} E_*^n = \frac{1}{2}\sum_j (f_*(x_i) - t_j^n)^2 \simeq R(f^*) &= (Y - F(f_*))^T (Y - F(f_*)) \\ &= \sum_i (y_i - f_*(x_i))^2 \simeq R^*(f_*) \\ &= (Y - F(f_*))^T (\hat{\tau}\mathcal{V} + \tau\mathcal{P})(Y - F(f_*)) \end{aligned} \tag{8.3.8}$$

where $\mathcal{V}$ and $\mathcal{P}$ are matrices and $\mathcal{P}$ is defined as

$$\mathcal{P} = \frac{1}{m}\sum_{s=1}^{m} \Phi_s \Phi_s^T$$

and $\Phi$ are statistical invariants functions defined on the training set $(x_i, y_i)$.

To modify the Neural Network method, we rewrite the problem of minimizing functional in the set of piecewise linear functions as the problem of minimizing the functional $E_*$ subject to constrains $x_i(k)$.

$$f\left(\sum w_{ji} X_j^n\right) \simeq x_i(k) = (g(W_1^T(k)x_i(k-1)), ..., g(W_{n_{(k-1)}}^T(k)x_i(k-1)))^T \tag{8.3.9}$$

In order to minimize functional $E_*^n$ subject to constrains $x_i(k)$, consider Lagrangian

$$L(W, X, B) = E_*^n + \sum_{i=1}^{l}\sum_{k=1}^{N-1} B_i(k)(x_i(k) - G([W(k)x_i(k-1)]))$$

where $B_i(k)$ are Lagrange multipliers. In order to find a minimum with respect to $W(k)$, the gradient descent procedure

$$W(k) \leftarrow W(k) - \lambda \frac{\partial L(W, X, B)}{\partial W(k)} \tag{8.3.10}$$

is used.

# A

# Montecarlo simulations-Numerical check

The purpose of the numerical experiments is to cross-validate and illustrate the theoretical results.

see suplementary material:
https://www.frontiersin.org/articles/10.3389/fnins.2022.971937/full

# Bibliography


[1] Nassim Abderrahmane, Edgar Lemaire, and Benoît Miramond. “Design Space Exploration of Hardware Spiking Neurons for Embedded Artificial Intelligence”. In: *Neural Networks* 121 (2020), pp. 366–386. ISSN: 18792782. DOI: 10.1016/j.neunet.2019.09.024. arXiv: 1910.01010. URL: https://doi.org/10.1016/j.neunet.2019.09.024.

[2] E D Adrian. *The Basis of Sensation*. Christophers, 1928.

[3] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

[4] Romain Brette. “Philosophy of the spike: rate-based vs. spike-based theories of the brain”. In: *Frontiers in systems neuroscience* (2015), p. 151.

[5] Iulia-Maria Comsa et al. “Temporal Coding in Spiking Neural Networks With Alpha Synaptic Function: Learning With Backpropagation”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021), pp. 1–14. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2021.3071976. URL: https://ieeexplore.ieee.org/document/9416238/.

[6] David Daniel Cox and Thomas Dean. “Neural networks and neuroscience-inspired computer vision”. In: *Current Biology* 24.18 (2014), R921–R929.

[7] Mike Davies et al. *Loihi: A Neuromorphic Manycore Processor with On-Chip Learning*. 2018. DOI: 10.1109/MM.2018.112130359.

[8] William Feller(1906-1970). *An Introduction to Probability Theory and Its Applications*. Wiley, 1950.

[9] Charlotte Frenkel. “Sparsity provides a competitive advantage”. In: *Nature Machine Intelligence* 3.9 (Sept. 2021), pp. 742–743. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00387-y. URL: http://dx.doi.org/10.1038/s42256-021-00387-y%20https://www.nature.com/articles/s42256-021-00387-y.

[10] Stephen B. Furber et al. “Sparse Distributed Memory Using Rank-Order Neural Codes”. In: *IEEE Transactions on Neural Networks* 18.3 (2007), pp. 648–659. DOI: 10.1109/TNN.2006.890804.

[11] Steve B Furber et al. “Sparse distributed memory using N-of-M codes”. In: *Neural Networks* 17.10 (2004), pp. 1437–1451.

[12] J Gautrais and S Thorpe. “Rate coding versus temporal order coding: a theoretical approach.” In: *Biosystems* 48.1-3 (1998), pp. 57–65.

[13] Jacques Gautrais. “Théorie et simulations d’un nouveau type de codage impulsionnel pour le traitement visuel rapide : le codage par l’ordre d’activation”. PhD thesis. Jan. 1997. DOI: 10.13140/RG.2.2.30052.50560.

[14] Wulfram Gerstner et al. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, 2014. DOI: 10.1017/CBO9781107447615.

[15] Tim Gollisch and Markus Meister. "Rapid neural coding in the retina with relative spike latencies." In: *Science* 319.5866 (Feb. 2008), pp. 1108–1111. DOI: 10.1126/science.1149639. URL: http://dx.doi.org/10.1126/science.1149639.

[16] J. Göltz et al. "Fast and energy-efficient neuromorphic deep learning with first-spike times". In: *Nature Machine Intelligence* 3.9 (Sept. 2021), pp. 823–835. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00388-x. arXiv: 1912.11443. URL: http://dx.doi.org/10.1038/s42256-021-00388-x%20https://www.nature.com/articles/s42256-021-00388-x.

[17] Wenzhe Guo et al. "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems". In: *Frontiers in Neuroscience* 15 (2021), p. 638474.

[18] Nitin Gupta and Mark Stopfer. "A Temporal Channel for Information in Sparse Sensory Coding". In: *Current Biology* 24.19 (2014), pp. 2247–2256. ISSN: 0960-9822. DOI: https://doi.org/10.1016/j.cub.2014.08.021. URL: https://www.sciencedirect.com/science/article/pii/S0960982214010343.

[19] Jakob Jordan et al. "Evolving interpretable plasticity for spiking networks". In: *eLife* 10 (Oct. 2021), pp. 1–33. ISSN: 2050-084X. DOI: 10.7554/eLife.66273. URL: https://elifesciences.org/articles/66273.

[20] Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988.

[21] Saeed Reza Kheradpisheh and Timothée Masquelier. "Temporal Backpropagation for Spiking Neural Networks with One Spike per Neuron". In: *International Journal of Neural Systems* 30.06 (June 2020), p. 2050027. ISSN: 0129-0657. DOI: 10.1142/S0129065720500276. URL: https://www.worldscientific.com/doi/abs/10.1142/S0129065720500276.

[22] Peter Kloppenburg and Martin Paul Nawrot. "Neural Coding: Sparse but On Time". In: *Current Biology* 24.19 (2014), R957–R959. ISSN: 0960-9822. DOI: https://doi.org/10.1016/j.cub.2014.08.041. URL: https://www.sciencedirect.com/science/article/pii/S0960982214010549.

[23] Teuvo Kohonen. "Correlation Matrix Memories". In: *IEEE Transactions on Computers* C-21.4 (1972), pp. 353–359. DOI: 10.1109/TC.1972.5008975.

[24] Mingxuan Liang, Jilin Zhang, and Hong Chen. "A 1.13$\mu$J/classification Spiking Neural Network Accelerator with a Single-spike Neuron Model and Sparse Weights". In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1. IEEE, May 2021, pp. 1–5. ISBN: 978-1-7281-9201-7. DOI: 10.1109/ISCAS51556.2021.9401607. URL: https://ieeexplore.ieee.org/document/9401607/.

[25] Wolfgang Maass. "Networks of spiking neurons: the third generation of neural network models". In: *Neural networks* 10.9 (1997), pp. 1659–1671.

[26] Henry Markram and Misha Tsodyks. "Redistribution of synaptic efficacy between neocortical pyramidal neurons". In: *Nature* 382.6594 (1996), pp. 807–810.

[27] Timothée Masquelier. “STDP Allows Close-to-Optimal Spatiotemporal Spike Pattern Detection by Single Coincidence Detector Neurons”. In: *Neuroscience* 389 (Oct. 2018), pp. 133–140. ISSN: 03064522. DOI: 10.1016/j.neuroscience.2017.06.032. URL: http://www.ncbi.nlm.nih.gov/pubmed/28668487%20https://linkinghub.elsevier.com/retrieve/pii/S0306452217304372.

[28] Timothée Masquelier and Saeed Reza Kheradpisheh. “Optimal Localist and Distributed Coding of Spatiotemporal Spike Patterns Through STDP and Coincidence Detection”. In: *Frontiers in Computational Neuroscience* 12.September (Sept. 2018), pp. 1–13. ISSN: 1662-5188. DOI: 10.3389/fncom.2018.00074. URL: https://www.frontiersin.org/article/10.3389/fncom.2018.00074/full.

[29] Carver Mead. “Neuromorphic Electronic Systems”. In: *Proceedings of the IEEE* (1990). ISSN: 15582256. DOI: 10.1109/5.58356.

[30] Adnan Mehonic and Anthony J Kenyon. “Brain-inspired computing needs a master plan”. In: *Nature* 604.7905 (2022), pp. 255–260.

[31] Maryam Mirsadeghi et al. “STiDi-BP: Spike Time Displacement based Error Back-Propagation in multilayer spiking neural networks”. In: *Neurocomputing* 427 (Dec. 2021), pp. 131–140. ISSN: 09252312. DOI: 10.1016/j.neucom.2020.11.052. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231220318452.

[32] Hesham Mostafa. “Supervised Learning Based on Temporal Coding in Spiking Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2017), pp. 1–9. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2017.2726060. arXiv: 1606.08165. URL: http://arxiv.org/abs/1606.08165%20http://ieeexplore.ieee.org/document/7999227/.

[33] Harideep Nair, John Paul Shen, and James E. Smith. “Direct CMOS Implementation of Neuromorphic Temporal Neural Networks for Sensory Processing”. In: (Aug. 2020), pp. 1–12. arXiv: 2009.00457. URL: http://arxiv.org/abs/2009.00457.

[34] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks”. In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63.

[35] Seongbin Oh et al. “Hardware Implementation of Spiking Neural Networks Using Time-To-First-Spike Encoding”. In: *arXiv* Vmm (June 2020), pp. 1–20. arXiv: 2006.05033. URL: http://arxiv.org/abs/2006.05033.

[36] Marc Osswald et al. “A spiking neural network model of 3D perception for event-based neuromorphic stereo vision systems”. In: *Scientific reports* 7.1 (2017), pp. 1–12.

[37] Seongsik Park et al. “T2FSNN: Deep Spiking Neural Networks with Time-to-first-spike Coding”. In: *arXiv* (Mar. 2020). arXiv: 2003.11741. URL: http://arxiv.org/abs/2003.11741.

[38] Seongsik Park et al. “T2FSNN: Deep spiking neural networks with time-to-first-spike coding”. In: *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2020, pp. 1–6.

[39] Laurent Perrinet, Manuel Samuelides, and Simon Thorpe. “Coding static natural images using spiking event times: do neurons cooperate?” In: *IEEE Transactions on neural networks* 15.5 (2004), pp. 1164–1175.

[40] Filip Ponulak and Andrzej Kasinski. “Introduction to spiking neural networks: Information processing, learning and applications.” In: *Acta neurobiologiae experimentalis* 71.4 (2011), pp. 409–433.

[41] BARRY J Richmond and LANCE M Optican. “Temporal encoding of two-dimensional patterns by single units in primate primary visual cortex. II. Information transmission”. In: *Journal of Neurophysiology* 64.2 (1990), pp. 370–380.

[42] Paul Rogister et al. “Asynchronous event-based binocular stereo matching”. In: *IEEE Transactions on Neural Networks and Learning Systems* 23.2 (2011), pp. 347–353.

[43] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. “Towards spike-based machine intelligence with neuromorphic computing”. In: *Nature* 575.7784 (2019), pp. 607–617.

[44] Bodo Rueckauer and Shih-Chii Liu. “Conversion of analog to spiking neural networks using sparse temporal coding”. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2018, pp. 1–5. ISBN: 978-1-5386-4881-0. DOI: 10.1109/ISCAS.2018.8351295. URL: https://ieeexplore.ieee.org/document/8351295/.

[45] Bodo Rueckauer and Shih-Chii Liu. “Conversion of analog to spiking neural networks using sparse temporal coding”. In: *2018 IEEE international symposium on circuits and systems (ISCAS)*. IEEE. 2018, pp. 1–5.

[46] Yusuke Sakemi et al. “A Supervised Learning Algorithm for Multilayer Spiking Neural Networks Based on Temporal Coding Toward Energy-Efficient VLSI Processor Design”. In: *arXiv* (Jan. 2020). arXiv: 2001.05348. URL: http://arxiv.org/abs/2001.05348.

[47] P Srivatsa et al. “You Only Spike Once : Improving Energy-Efficient Neuromorphic Inference to ANN-Level Accuracy”. In: (2020).

[48] P Suppes, M Pavel, and J C Falmagne. “Representations and Models in Psychology”. In: *Annual Review of Psychology* 45.1 (1994), pp. 517–544. DOI: 10.1146/annurev.ps.45.020194.002505. eprint: https://doi.org/10.1146/annurev.ps.45.020194.002505. URL: https://doi.org/10.1146/annurev.ps.45.020194.002505.

[49] S J Thorpe and J Gautrais. “Rank Order Coding”. In: *Computational Neuroscience : Trends in Research*. Ed. by James M Bower. New York: Plenum Press, 1998, pp. 113–118.

[50] S J Thorpe and M Imbert. “Biological constraints on connectionist modelling”. In: *Connectionism in perspective*. Ed. by R Pfeifer et al. Amsterdam: Elsevier, 1989, pp. 63–92.

[51] Simon Thorpe, Arnaud Delorme, and Rufin Van Rullen. “Spike-based strategies for rapid processing”. In: *Neural networks* 14.6-7 (2001), pp. 715–725.

[52] Simon Thorpe, Denis Fize, and Catherine Marlot. “Speed of processing in the human visual system”. In: *nature* 381.6582 (1996), pp. 520–522.

[53] Martin J Tovee and Edmund T Rolls. “Information encoding in short firing rate epochs by single neurons in the primate temporal visual cortex”. In: *Visual cognition* 2.1 (1995), pp. 35–58.

[54] Vladimir Vapnik and Rauf Izmailov. “Complete statistical theory of learning: learning using statistical invariants”. In: *Conformal and Probabilistic Prediction and Applications*. PMLR. 2020, pp. 4–40.

[55] Tom Verhoeff. “Delay-insensitive codes—an overview”. In: *Distributed computing* 3.1 (1988), pp. 1–8.

[56] Wikipedia. *Sensitivity index wikipedia*. URL: https://en.wikipedia.org/wiki/Sensitivity_index (visited on 12/13/2021).

[57] Malu Zhang et al. “Rectified Linear Postsynaptic Potential Function for Backpropagation in Deep Spiking Neural Networks”. In: *arXiv* (2020), pp. 1–11. arXiv: 2003.11837. URL: http://arxiv.org/abs/2003.11837.

[58] Shibo Zhou et al. “Temporal-Coded Deep Spiking Neural Network with Easy Training and Robust Performance”. In: (2019). arXiv: 1909.10837. URL: http://arxiv.org/abs/1909.10837.