Using event-based metric for event-based neural network weight adjustment

Bruno Cessac, Rodrigo Salas, Thierry Viéville

Using event-based metric for event-based neural network weight adjustment

Bruno Cessac[2] and Rodrigo Salas[3] and Thierry Viéville[1] *

(1) Inria Cortex, France http://cortex.loria.fr
(2) Inria Neuromathcomp, France
(3) Universidad de Valparaiso. Departamento de Ingenieria Biomedica. Chile

Abstract. The problem of adjusting the parameters of an event-based network model is addressed here at the programmatic level. Considering temporal processing, the goal is to adjust the network units weights so that the outcoming events correspond to what is desired. The present work proposes, in the deterministic and discrete case, a way to adapt usual alignment metrics in order to derive suitable adjustment rules. At the numerical level, the stability and unbiasness of the method is verified.

(Extended version of the ESANN'12 accepted publication)

1 Introduction

Studying the computational power of neural networks with event-based activity (e.g.: [1, 2]) is a well-addressed topic, see [3, 4] for a recent review about spiking network computation, while [5] provides a detailed discussion on temporal aspects of such computations. See [6] for further details on the related modeling choices. In order to contribute to this general topic, we develop here a framework allowing us to effectively adjust the network parameters in order to tune the outcoming events.

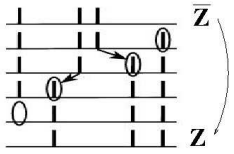
Position of the problem We consider an input/output dynamical system with N units, governed by a recurrent function, $\mathbf{V} = \{\cdots V_n[t] \cdots\}$ being the *output* state variable value of the units of output index $n \in \{0, N\{$ (i.e., $0 \leq n < N$) at time $t \in \{0, T\{$. Some output units may be "hidden", i.e. not observed. Here $\mathbf{W}$ stands for the network parameters or "weights", to be tuned. The exact form of $\mathbf{V}$ is not relevant at this stage, but the gradient $\nabla_{\mathbf{W}}\mathbf{V}_{\mathbf{W}}$ must be well-defined in order to adjust $\mathbf{W}$. One track is to consider regular forms of $\mathbf{V}$. For a spiking neuron network this means that we have to consider either Hodgkin-Huxley equations, or some suitable reduction like the FitzHugh-Nagumo model or the SRM model [2]. Another track, is to "mollify" $\mathbf{V}$, i.e., defines it as the limit of a series of regular functions, as experimented in [7].

We define an event $Z_n[t] \stackrel{\text{def}}{=} H(V_n[t] - \theta) \in \{0, 1\}$, where H is the Heaviside function, as the fact that the output value is higher than a threshold θ. The

*Partially supported by the ANR KEOpS project and CORTINA associated team.

goal is thus to adjust the output events $\mathbf{Z}$ of the deterministic discrete-time dynamical system $\mathbf{V}$, with respect to a reference output events $\bar{\mathbf{Z}}$. The key point is to deal with the fact that the notion of event is intrinsically "discontinuous".

Considering alignment metric We define the distance between two finite event-trains $\bar{\mathbf{Z}}$, $\mathbf{Z}$ as *the minimum cost of transforming one event-train into another.* See [8, 9] for a general introduction. Following [5], we consider a generalized alignment metric: non-stationary cost (e.g., recent events may count more than older ones) and non-linear shift (e.g., neglecting tiny delays), as described in Fig. 1. Two kinds of operations are defined for an alignment metric.
(i) Event insertion/deletion, the cost of each operation being set to $\gamma_{\bar{t}}^{\pm}$ at time $\bar{t}$, e.g., $\gamma_{\bar{t}}^{\pm} = 1$, while non-stationary different insertion/deletion costs may be defined.
(ii) Event shift, the cost to shift from one event in $\bar{\mathbf{Z}}$ at time $\bar{t}$ to one event $\mathbf{Z}$ at time t, being an increasing positive function of the non-stationary normalized shift delay $\phi_{\bar{t}}((\bar{t}-t)/\tau)$, for a given time-constant τ (e.g. $\phi_{\bar{t}}((\bar{t}-t)/\tau) = |\bar{t}-t|/\tau$), while non-stationary non-linear different forward/backward shift-cost may be defined, since $\phi()$ is parameterized by $\bar{t}$.



From the upper to the lower event train is shown, using from top to bottom an insertion, a rightward shift, a leftward shift and a deletion respectively

Fig. 1: An example of minimal alignment (borrowed from [8]).

Obviously the distance is zero (no editing operation) if and only if both trains are equal, is always bounded by the number of events in both event-trains (i.e. the cost of deleting/inserting all events), thus also by twice the number of samples in the discretized case. For small τ, the distance approaches the number of non-coincident events, since instead of shifting events it is cheaper to insert/delete non-coincident events, while when $\tau \to 0, \gamma_{\bar{t}}^{\pm} = 1$ we obtain the coincidence (or Hamming) distance equal to the number of non-coincident events. Given two time sequences with the same number of events, there is always a τ high enough for the distance to correspond to the weighted sum of time differences between both train events, as used in, e.g., [4]. More generally, for high τ, the distance basically equals the difference in event number (rate distance) [8].
When considering event-trains with more than one unit, our approach consists to sum the distances for each unit alignment, i.e., consider each unit independently, avoiding the related estimation to suffer from NP-completeness [9].

2 Defining indexed alignment divergence

Since we want to tune the $\mathbf{Z}$ events in order to approximate the $\bar{\mathbf{Z}}$ events, let us introduce an alignment indexation as follow: $\delta(t) = \bar{t} - t$ if the two events $Z[t] = 1$ and $\bar{Z}[\bar{t}] = 1$ are aligned by a shift, $\delta(t) = \pm 0$ to code for an insertion/deletion, while $\delta(t) = 0$ otherwise. In words, we not only compute the distance but make explicit the alignment operations ($shift$, $deletion$, $insertion$) allowing to "edit" $\mathbf{Z}$ in order to obtain $\bar{\mathbf{Z}}$. The δ code function is used to explicitly match both trains.

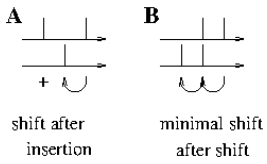
The distance $d_{\bar{k},k}$ between the first $\bar{k}$ events in $\bar{\mathbf{Z}}$ and the first k events $\mathbf{Z}$ and the related δ indexing are iteratively defined by induction, after [8] but now generalized (see [7] for a detailed derivation). We write $t_k, k > 0$ the k-th value such that $Z[t_k] = 1$, with a similar notation for $\bar{t}_{\bar{k}}, \bar{k} > 0$. On one hand, $d_{\bar{k},0} = \sum_{\bar{l}<\bar{k}} \gamma^-_{\bar{t}_{\bar{l}}}$, since the distance between any event-train and the empty event-train corresponds to the cost of deleting all events, while $\delta(t_{\bar{k}}) = -0$ in this case. Similarly, $d_{0,k} = \sum_{l<k} \gamma^+_{t_l}$ corresponds to inserting all events, with $\delta(t_{\bar{k}}) = +0$. On the other hand:

$$
d_{\bar{k}+1,k+1} = \\
\min \begin{cases}
d_{\bar{k},k+1} & + & \gamma^-_{\bar{t}_{\bar{k}}} & (deletion), & \delta(t_{k+1}) & = & -0 \\
d_{\bar{k}+1,k} & + & \gamma^+_{t_k} & (insertion), & \delta(t_k) & = & +0 \\
d_{\bar{k},k} & + & \phi_{\bar{t}_{\bar{k}}}\left(\frac{\bar{t}_{\bar{k}} - t_k}{\tau}\right) & (shift), & \delta(t_k) & = & \bar{t}_k - t_k.
\end{cases} \tag{1}
$$

Obviously, several alignment operation sequences may lead to the same minimal alignment cost. In order to make a choice, from the last time to the previous time, we consider that shift is preferable to insertion/deletion, since it is a reasonable assumption to heuristic that it is going to have a less important influence on the dynamics than the apparition/cancellation of an unexpected event. This defines algorithmically a unique well-defined indexing function for a given distance, as illustrated in Fig. 2, thus solves the ambiguities. On the reverse, solving these ambiguities allows us to define algorithmically a unique indexing function.

Although computing such a distance and indexes seems subject to a combinatorial complexity, this is a quadratic algorithm (i.e. with a complexity equal to the product of the numbers of events), and its derivation, done by induction, is similar to usual alignment distance calculations [9, 5]. Regarding indexing, this means that we do not have to explore all possible alignment operation sequences, in order to define a globally well-defined process.

This indexing definition also allows us to enrich the original alignment distance by not only considering a composite number describing the distance in terms of shift and insertion/deletion, but allowing to make explicit a numerical approximation of the number of shifts *versus* insertion/deletion. This is the same feature as in message-passing alignment mechanisms [10], but defined here in a much simpler context.



Edition is realized from the most recent event to the oldest event in the past, i.e., from right to left. The previous specification solves the ambiguity. **A** From right to left, shift is preferred to insertion, thus shift precedes insertion. **B** From right to left, two minimal shifts are preferred to a higher shift and a event coincidence.

Fig. 2: Solving ambiguous equal distance alignments, in the case where a shift cost equals the insertion/deletion cost.

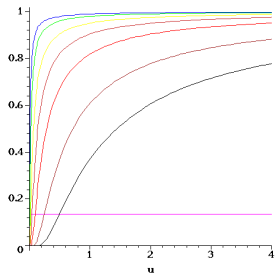
3 Mollified version of the alignment distance

From the previous construction we now introduce the key idea of the paper, i.e., propose a variational expression of the alignment distance. To this aim, we "mollify" the event generation mechanism, i.e. replace the Heaviside function by a suitable convolution $H_\upsilon = \upsilon * H = H(u+\sqrt{\upsilon}\,\nu)\,\exp(-\upsilon/(u+\sqrt{\upsilon}\,\nu))$ where ν is a *margin* maintaining the state at a non-infinitesimal distance to the threshold, and $\upsilon \to 0$ is the mollification factor (see [6] for details)), as show in Fig. 3. We obtain after some algebra given in [6]: $d\left(\bar{\mathbf{Z}}, \mathbf{Z}\right) = \lim_{\upsilon\to 0} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right)$, with:

$$\begin{aligned} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right) &= \sum_{nt,\delta[t]=0} \gamma_t^{\pm}\, H_\upsilon\left(\left(1-2\,\bar{Z}_n[t]\right)\left(V_n[t]-\theta\right)\right) \\ &+ \sum_{nt,\delta[t]\neq 0} \phi_t\left(\tfrac{\delta[t]}{\tau}\right) \left(\begin{array}{l} Z_n[t]+ \\ \left(1-Z_n[t]\right) H_\upsilon\left(-\frac{\delta[t]}{\tau}\, H_\upsilon(\theta - V_n[t])_{\nu=0}\right) \end{array} \right) \end{aligned} \tag{2}$$

The key point is that now the criterion is not defined with respect to $\mathbf{Z}$ but $\mathbf{V}$. Qualitatively an increase of $V_n[t]$ tends to shift event in the past, avoid deletion but induce insertion of event, whereas a decrease of $V_n[t]$ tends to shift event in the future, induce deletion but avoid insertion of event. Changes are now differentiable, thanks to the mollification and the gradient $\nabla_{\mathbf{W}} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right) = \nabla_{\mathbf{V}} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right) \nabla_{\mathbf{W}}\mathbf{V}$ is obvious to derive. We thus can *tune* $\mathbf{V}$, thus $\mathbf{W}$ to optimize the alignment metric as desired, with a straight-forward implementation for a feed-forward system and the need of specific method for a recurrent structure, as developed elsewhere [7]. The metric allows us to calculate the appropriate network weights by a simple numerical minimization.

Since H_υ is convex for suitable υ [6], for a fixed value of δ the criterion is convex as the sum of positively weighted functions. However the criterion is also optimized with respect to δ and as soon as an event occurrence is modified by a variation of the weights, the indexing is to be recalculated, while, up to our best knowledge, there is no chance to guarantee a global minimum, so that we now turn to numerical verification.



It is drawn here for $\nu = 0$ and in *black*, *brown*, *red*, *orange*, *yellow*, *green*, *blue*, for $\upsilon = [1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01]$, respectively. The curves are convex below the magenta horizontal line.

Fig. 3: Defining the mollification of the Heaviside function $H()$.

4 Numerical experiment

Criterion calibration. In order to estimate the performance of the estimation, we must first quantify to which extends we obtain numbers "better than by chance", i.e., to which extends the minimized alignment distance yields a better result than if the output would have been random (a small distance may simply mean that events are sparse !). In order to obtain a correct order of magnitude, we considered the *normalized alignment distance* with respect to a random event train of the same rate. More precisely, if two event trains of T samples are drawn from a Bernoulli distribution of rates r (i.e., samples are random and independent), it is straightforward to obtain the average coincidence distance, i.e. $E[d(\bar{\mathbf{Z}}, \mathbf{Z})] = 2\,T\,r\,(1-r)$. However, the same derivation is not obvious for an alignment distance parameterized with τ and we have numerically interpolated the value as a power of τ, for the standard alignment distance, obtaining for $\gamma_t^{\pm} = 1, \phi_t(s) = |s|$, with a residual standard deviation better than 1.5%:

$$E[d(\bar{\mathbf{Z}}, \mathbf{Z})] = 2\,T\,r\,(1-r)\,\frac{1.183 + 0.183\,r\,(1-r)}{(\tau+1)^{0.265+1.444\,r\,(1-r)}}.$$

Numerical robustness. Let us now illustrate the previous developments considering a leaky integrate and fire (LIF) network, as in [11]. As a test, we have generated hundred of input/output data sets using a "master" network and have verified that the learning algorithm applied on another model of the same dimensions is able to find weights that reproduce the input/output function. Weights values are randomly drawn from a Gaussian distribution of zero mean and standard-deviation $\sigma \in [0.1, 10]$. The LIF resetting mechanism is mollified as for the event thresholding.
This is a basic verification of both the correctness of the code and the numerical stability of the estimation. Hundred of runs have provided correct results, as expected. For long length $T > 10^5$ and complex dynamics the method may fail finding the exact solution with the standard parameters. For small length epoch, as expected, there is always an exact solution, in fact there is one, even if the

raster is not generated from the same model [11].
These tests have been done for various values of $\tau \geq 0$ and using several values of margin $\nu \in [0.01, 0.1]$. The key point is that we can obtain good numerical results "even if" profiles are finally very sharp, using the proposed continuation method, consisting of numerically drive $\upsilon \to 0$. We have experimented using the conjugate gradient algorithm of the GSL (`http://www.gnu.org/s/gsl`) library, but have also checked that this is not a critical choice. Robustness has been checked for $\gamma_t^{\pm} = 1, \phi_t(s) = |s|$ and several for generalized metric also. Further numerical results are provided as supplementary material of this submission, while the code is available in the open-source EnaS library (`http://enas.gforge.inria.fr`).

5 Conclusion

The key point, here, is the non-learnability of even-based networks [12], since it is proved that this problem is NP-complete, when considering the estimation of both weights in the general case, except for exact simulation [11]. We show that we can "elude" this caveat and propose an alternate efficient estimation mechanism, inspired by alignment metrics used in spike train analysis [9], thus providing a complement of other estimation approaches [4], beyond usual convolution metric [9, 5]. At last, the proposed mollification *is* a series of convolution metric, but that converges towards the expected alignment metric.

References

[1] F. Rieke, D. Warland, Rob de Ruyter van Steveninck, and William Bialek. *Spikes, Exploring the Neural Code*. The M.I.T. Press, 1996.

[2] W. Gerstner and W. Kistler. *Spiking Neuron Models*. Cambridge University Press, 2002.

[3] H. Paugam-Moisy and S.M. Bohte. *Handbook of Natural Computing*, chapter Computing with Spiking Neuron Networks. Springer Verlag, 2009.

[4] Benjamin Schrauwen. *Towards Applicable Spiking Neural Networks*. PhD thesis, Universiteit Gent, Belgium, 2007.

[5] Bruno Cessac, H. Paugam-Moisy, and Thierry Viéville. Overview of facts and issues about neural coding by spikes. *J. Physiol. Paris*, 104:5–18, 2010.

[6] T. Viéville, B. Cessac, and R. Salas. Using event-based metric for network weight adjustment (extended version). Supplementary material for ESANN'12 http://www.loria.fr/ vthierry/results/vieville-etal-2012b.pdf, INRIA, 2011.

[7] T. Viéville and L. Bougrain. A general algorithm to estimate recurrent networks weights (extended version). Supplementary material for CAP'12 http://www.loria.fr/ vthierry/results/vieville-etal-2012a.pdf, INRIA, 2011.

[8] J.D. Victor. Spike train metrics. *Current Opinion in Neurobiology*, 15(5):585–592, 2005.

[9] C. Houghton and Victor. *Measuring representational distances - the spike-train metrics approach*. MIT Press,, 2011.

[10] J. Dauwels, F. Vialatte, T. Rutkowski, and A. Cichocki. Measuring neural synchrony by message passing. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[11] H. Rostro-Gonzalez, Bruno Cessac, Juan-Carlos Vasquez, and Thierry Viéville. Back-engineering of spiking neural networks parameters. Research report, INRIA, 2010. in preparation.

[12] Jiří Šíma and Jiří Sgall. On the nonlearnability of a single spiking neuron. *Neural Computation*, 17(12):2635–2647, 2005.

[13] S. Chemla, F. Chavane, T. Vieville, and P. Kornprobst. Biophysical cortical column model for optical signal analysis. In W. R. Holmes, R. Jung, and F. Skinner, editors, *Sixteenth Annual Computational Neuroscience Meeting (CNS)*, volume 8, Suppl 2 of *BMC Neuroscience*, July 2007.

[14] Thierry Viéville, S. Chemla, and P. Kornprobst. How do high-level specifications of the brain relate to variational approaches? *J. Physiol. Paris*, 101, 2007.

[15] N. Rougier. Dynamic neural field with local inhibition. *Biological Cybernetics*, 94(3):169–179, 2006.

[16] Bruno Cessac. A discrete time neural network model with spiking neurons. rigorous results on the spontaneous dynamics. *J. Math. Biol.*, 56(3):311–345, 2008.

[17] Bruno Cessac and T. Viéville. On dynamics of integrate-and-fire neural networks with adaptive conductances. *Frontiers in neuroscience*, 2(2), July 2008.

[18] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris Jr., M. Zirpe, T. Natschläger, D. Pecevski, G. Bard Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, Thierry Viéville, E. Muller, A. P. Davison, S. El Boustani, and Alain Destexhe. Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of Computational Neuroscience*, 23(3):349–398, 2007.

[19] R. Guyonneau, R. vanRullen, and S.J. Thorpe. Neurons tune to the earliest spikes through stdp. *Neural Computation*, 17:859–879, 2005.

[20] A. Delorme, L. Perrinet, and S. Thorpe. Network of integrate-and-fire neurons using rank order coding b: spike timing dependant plasticity and emergence of orientation selectivity. *Neurocomputing*, 38:539–545, 2001.

[21] P. Baudot. *Nature is the code: high temporal precision and low noise in V1.* PhD thesis, Univ. Paris 6, 2007.

[22] Richard H. Masland and Paul R. Martin. The unsolved mystery of vision. *Curr Biol.*, 17(15):R577–82, Aug 2007.

[23] K. Koch, J. McLean, M. Berry II, P. Sterling, V. Balasubramanian, and M.A. Freed. Efficiency of information transmission by retinal ganglion cells. *Current Biology*, 14(17):1523–1530, 2004.

[24] C. VanVreeswijk. *What is the neural code?* 23 Problems in System neuroscience. van Hemmen, J.L. and Sejnowski, T.Jr. (eds), Oxford University Press, 2004.

[25] Jean A. Dieudonne. *Treatise on Analysis Volume II.* Academic Press, 1976.

[26] R. Benedetti and J-J. Risler. *Real algebraic and semi-algebraic sets.* Hermann, Paris, 1990.

[27] W. Maass. On the relevance of time in neural computation and learning. *Theoretical Computer Science*, 261:157–178, 2001. (extended version of '97, in LNAI 1316:364-384).

[28] P. Dayan and L.F. Abbott. *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems.* MIT Press, 2001.

[29] Anton Maximilian Schäfer and Hans Georg Zimmermann. Recurrent neural networks are universal approximators. *Lecture Notes in Computer Science*, 4131:632–640, 2006.

[30] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[31] W. Maass. Fast sigmoidal networks via spiking neurons. *Neural Computation*, 9:279–304, 1997.

[32] W. Maass and T. Natschlager. Networks of spiking neurons can emulate arbitrary hopfield nets in temporal coding. *Neural Systems*, 8(4):355–372, 1997.

[33] Wolfgang Maass and Christopher M. Bishop, editors. *Pulsed Neural Networks*. MIT Press, 2003.

[34] E.M. Izhikevich. Polychronization: Computation with spikes. *Neural Computation*, 18:245–282, 2006.

[35] Hélène Paugam-Moisy, Régis Martinez, and Samy Bengio. Delay learning and polychronization for reservoir computing. *Neurocomputing*, 71:1143–1158, 2008.

A On modeling choices regarding deterministic discrete-time models.

At the microscopic biological level, units are punctual models of neurons, events are action potentials, weights stand for synaptic strength, and the dynamical system is the neural network under consideration. At the mesoscopic biological level, units are cortical columns (see, e.g., [13] for a discussion on the concept), events are synchronization, rhythms, or sudden activity change, weights are related to the average connection strength, and the dynamical system is a cortical map (see e.g. [14] for a discussion on the concept), including dynamical neural fields (see, e.g., [15]).

Considering discretized temporal mapping.

Considering that time is discretized is a twofold issue. On one hand, it corresponds to the fact that not all continuous time sequences correspond to event trains, since they are constrained by the network dynamics, yielding global time constraints such as the fact that inter-event intervals are bounded by a refractory period r and the fact that event times are defined up to some absolute precision δt (see [5] for a detailed discussion), this being true in both biological and electronical implementations. The maximal amount of information, for one unit, is thus bounded during a finite period $[0, T]$ as stated in [5]:

$$\frac{T}{r} \log_2 \left(\frac{T}{\delta t} \right) bits.$$

In a biological context, the order of magnitude is of $1K$ $bits/second$ for a neuron, in coherence with biological observations [1]. On the other hand, at a pragmatic level, time discretized network models are rather easy to study theoretically [16, 17], trivial to simulate (contrary to continuous time models, see [18] for discussion), and correspond without bias to what happens in a computer. We thus focus on discrete time and are going to briefly point out, from step to step, to which extent the present development can be applied to continuous time frameworks.

Considering a deterministic framework.

We also consider here a deterministic framework, i.e. work in a context where not the "average" input/output response, but an exact or approximate *specific* input/output response is targeted (i.e. in a context where "each spike may

matter" [19, 20], which seems to be the case, e.g., in the biological visual system working in natural scenario [21, 22, 23]). More precisely, from [5], we propose the following pragmatic view of the network result coding scheme (i.e. the "neural code" in a biological context [24, 1]): two results correspond approximately to the same code if their *distance* with respect to a given metric or pseudo-metric[1] is small. For instance, considering a rank coding scheme (i.e. in a context where the *relative temporal sorting* of the events matter, but not their exact temporal values [20]) the related pseudo-metric is discrete and easy to state: network output are equivalent if the ranks of event trains match, and not-equivalent otherwise. Contrary to this binary choice, our proposal is to introduce a richer structure: The proposed modeling view is not only to consider a weak notion of network coding where two codes can only be either equal or different, but a more general notion where two codes are similar up to some quantified distance. This seems to correspond to a more realistic view of, for instance, the still mysterious "neural code" (see [5] for a discussion), and at the estimation level allows variational optimization mechanisms to be used.

B Variational formulation of the alignment metric

Let us derive and explain the choice of (2). We proceed in three steps.

Coincidence metric mollification. We start with the simple case where $\tau = 0$. In that case we have no shift but only insertion/deletion. We count γ_t^+ if we have to insert a spike ($Z_n[t] = 0$ and $\bar{Z}_n[t] = 1$) and count γ_t^- if we have to delete a spike ($Z_n[t] = 1$ and $\bar{Z}_n[t] = 0$). The total counts writes:

$$\begin{array}{rcl} d_0\left(\bar{\mathbf{Z}}, \mathbf{V}\right)\big|_{\tau=0} & \stackrel{\text{def}}{=} & \sum_{nt} \gamma_t^{\pm} \left|\bar{Z}_n[t] - Z_n[t]\right| \\ & = & \sum_{nt} \gamma_t^{\pm} H\left(\left(1 - 2\,\bar{Z}_n[t]\right)\left(V_n[t] - \theta\right)\right), \end{array} \tag{3}$$

writing:

$$\gamma_t^{\pm} \stackrel{\text{def}}{=} \left\{ \begin{array}{rcl} \gamma_t^+ & , & \bar{Z}_n[t] = 1, \delta[t] = 0 \\ \gamma_t^- & , & \bar{Z}_n[t] = 0, \delta[t] = 0 \\ 0 & , & \delta[t] \neq 0. \end{array} \right. \tag{4}$$

This comes from the fact that $|\bar{Z}_n[t] - Z_n[t]| \in \{0, 1\}$ is equal to 1 if an only if $\bar{Z}_n[t] \neq Z_n[t]$, while the second form derives from the fact, obvious to derive,

[1] A *metric*, corresponding to the intuitive notion of "distance" $d(\mathbf{R}, \mathbf{R}')$ between two results $\mathbf{R}$ and $\mathbf{R}'$, is a real function, *positive* ($d(\mathbf{R}, \mathbf{R}') \geq 0$), *symmetric* ($d(\mathbf{R}, \mathbf{R}') = d(\mathbf{R}', \mathbf{R})$), *definite* ($d(\mathbf{R}, \mathbf{R}') = 0 \Leftrightarrow \mathbf{R} = \mathbf{R}'$))), thus *semi-definite* ($d(\mathbf{R}, \mathbf{R}) = 0$), and *subadditive* ($d(\mathbf{R}, \mathbf{R}') \leq d(\mathbf{R}, \mathbf{R}'') + d(\mathbf{R}'', \mathbf{R}')$, given any third result $\mathbf{R}''$).
If such a function is positive, definite, subadditive, but not symmetric, it is called a *divergence*. If such a function is positive, symmetric, subadditive, but only semi-definite while not definite, it is called a *pseudo-distance*. Thanks to the subadditivity, a pseudo-distance induces a distance on the quotient space of the *equivalence classes* of elements at a zero distance to of each-others. On the reverse, an equivalence relation $\equiv$, corresponds to a *discrete* pseudo-metric ($d(\mathbf{R}, \mathbf{R}') \in \{0, 1\}$) such that $\mathbf{R} \equiv \mathbf{R}' \Leftrightarrow d(\mathbf{R}, \mathbf{R}') = 0$. It is the coarser pseudo-metric compatible with the related equivalence classes, while finer distances quantify to which extents two results differ (see, e.g., [25] for a text book reference).

that:

$$|\bar{Z}_n[t] - Z_n[t]| = |\bar{Z}_n[t] - H\left(V_n[t] - \theta\right)| = H\left(\left(1 - 2\,\bar{Z}_n[t]\right)\left(V_n[t] - \theta\right)\right).$$

If $\gamma_t^{\pm} = 1$ the distance is the Hamming distance, counting the number of non-coincidences.

This distance is discontinuous, with a jump each time $V_n[t]$ crosses the threshold θ. We now replace the Heaviside function by a mollification, H_υ in (3). The exact choice of H_υ is detailed in section C. The corresponding mollified metric $d_\upsilon(\bar{\mathbf{Z}}, \mathbf{V})$, is now explicitly varying with $\mathbf{V}$, not only the value of $\mathbf{Z}$, allowing to tune the state value in order to it drive towards the correct side of the threshold.

The term $H_\upsilon\left(\left(1 - 2\,\bar{Z}_n[t]\right)\left(V_n[t] - \theta\right)\right)$ now increases the value of $V_n[t]$ if $\bar{Z}_n[t] = 1$ and decreases the value of $V_n[t]$ if $\bar{Z}_n[t] = 0$ and, thus drive the the state value towards the correct threshold side.

The interest of using the second form of (3), is that it has a symmetric effect for values below and above the threshold, which is not the case with the former form.

Variational form of the alignment metric. We now consider an alignment metric with its corresponding indexing δ.

As for the distance calculation or the divergence indexation, we can construct the variational form by double induction, on the n-th value such that $Z_n[t_n] = 1$ and the $\bar{n}$-th value such that $\bar{Z}_n[\bar{t}_{\bar{n}}] = 1$. This is due to the fact that, in a minimal path, each event can be either deleted or shifted once to coincide with an event in the other event-train. Also, an event can be inserted only at a time that matches the occurrence of an event in the other event-train.

Furthermore, a minimal path cannot include an insertion of an event that is later deleted or shifted, or a deletion of an event that is later inserted or shifted, or a shift in both direction, since the cost of such path can be reduced by eliminating some steps. Individual events cannot intersect, since uncrossing them reduces the amount of shifting. This means that $t + \delta[t]$ is an increasing function, i.e.,:

$$\begin{aligned} \forall t' < t, \bar{Z}_n[t' + \delta[t']] = \bar{Z}_n[t + \delta[t]] = 1 \quad &\Rightarrow \quad t' + \delta[t'] < t + \delta[t] \\ &\Leftrightarrow \quad -1 < \tfrac{\delta[t] - \delta[t']}{t - t'}. \end{aligned} \tag{5}$$

We write $\{t_1, t_2, \cdots\}$, i.e., $t_k, k > 0$ the k-th value such that $Z[t_k] = 1$, with a similar notation for $\bar{t}_{\bar{k}}, \bar{k} > 0$.

We start with $d_{\bar{k},0} = \sum_{l<k} \gamma^-_{\bar{t}_{\bar{k}}}$ due to the fact that the distance between any event-train and the empty event-train corresponds to the cost of deleting all events, while $\delta(t_{\bar{k}}) = -0$ in this case. Similarly, $d_{0,k} = \sum_{l<k} \gamma^+_{t_k}$ corresponds to inserting all events, with $\delta(t_{\bar{k}}) = +0$ in this case.

In order to calculate $d_{\bar{k}+1,k+1}$ we may either perform a deletion from $d_{\bar{k},k+1}$, an insertion from $d_{\bar{k}+1,n}$ or a shift from $d_{\bar{k},n}$ as made explicit in (1).

- If there is deletion, we obtain $d_{\bar{k}+1,k+1} = d_{\bar{k},k+1} + \gamma^-_{\bar{t}_{\bar{k}}}$, while $Z_n[t_{k+1}] = 0$ and $\bar{Z}_n[\bar{t}_{\bar{k}}] = 1$. And we must have $Z_n[\bar{t}_{\bar{k}}] = 0$ also, otherwise it would be costless not to perform the deletion. The cost to take into account is $\gamma^-_{\bar{t}_{\bar{k}}}$ since the event deletion occurs at time $\bar{t}_{\bar{k}}$.

- If there is an insertion, in a similar way, $d_{\bar{k}+1,k+1} = d_{\bar{k}+1,n} + \gamma^+_{t_k}$, while $Z_n[t_k] = 1$ and $\bar{Z}_n[t_k] = 0$.

- If there is shift, we obtain $d_{\bar{k}+1,k+1} = d_{\bar{k},k} + \phi_{t_k}\left(\frac{\bar{t}_{\bar{k}} - t_k}{\tau}\right)$, with $Z_n[t_k] = 1$ and $\bar{Z}_n[\bar{t}_k] = 1$, while we must have $\bar{Z}_n[t_k] = 0$ (otherwise a trivla better alignment occurs at t_k).

All together, we obtain the induction given in (1). Now if we want to derive a form similar to (3) we must recalculate, given a function δ, i.e., given the choices between insertion/deletion and shifts being fixed, the alignment function, now from an induction over the time t. To this aim, we write $d_{\{0,t\{} = d\left(\bar{\mathbf{Z}}, \mathbf{Z}\right)\big|_{0 \le t' < t}$. For $t = 0$, obviously $d_{\{0,0\{} = 0$. Assuming we have calculated $d_{\{0,t\{}$ at time t, we may have the following cases:

$\left(Z_n[t], \bar{Z}_n[t]\right)$	$d_{\{0,t+1\{} - d_{\{0,t\{}$	
$(0,0)$	0	no event
$(1,0)$	γ^+_t	insertion of $\bar{Z}_n[t]$
$(1,0)$	ϕ_t	shift source from to $Z_n[t]$ to $\bar{Z}_n[t + \delta[t]]$
$(0,1)$	γ^-_t	deletion of $\bar{Z}_n[t]$
$(0,1)$	0	shift target at $\bar{Z}_n[t]$ from $Z_n[t'], t = t' + \delta[t']$
$(1,1)$	0	coincidence
$(1,1)$	ϕ_t	shift at $\bar{Z}_n[t]$ from $Z_n[t'], t = t' + \delta[t']$ and from $\bar{Z}_n[t + \delta[t]]$ to $Z_n[t]$

where $\phi_t \stackrel{\text{def}}{=} \phi_t\left(\frac{\delta[t]}{\tau}\right)$. The key point is that we claim these are the only possible cases:
- when $(0,0)$ nothing happens,
- when $(1,0)$ the event $Z_n[t]$ can only be inserted or the source of a shift,
- when $(0,1)$ the event $\bar{Z}_n[t]$ can only be deleted or the target of a shift,
- when $(1,1)$ it may either be a coincidence of both events, or the result of two shifts, other operations being not optimal as discussed before.

Furthermore, since the shift cost must be counted only once and is taken at the time $\bar{t}$, it must be counted at the source but not the target of the shift.

All this can be summarized writing:

$$d_{\{0,t+1\{} = d_{\{0,t\{} + \gamma^{\pm}_t \left|\bar{Z}_n[t] - Z_n[t]\right| + \phi_t$$

with $\gamma^{\pm}_t$ defined from (4) and it is obvious to verify that the formula is true for the seven possible cases.

As a consequence, for a event train $\bar{Z}_n[t], t \in \{0, T\{$ and given an optimal indexing δ, we can write the corresponding alignment metric as $d\left(\bar{\mathbf{Z}}, \mathbf{Z}\right) = \sum_n d_{\{0,T\{}$, using the previous formula.

On the reverse, consider $d_{\{0,T\{}$ as a function of the indexing δ, the alignment

distance corresponds to the minimal value among all indexing, i.e. we can write[2]:

$$\delta = \text{argmin}_\delta d_{\{0,T\{} \quad , \quad d_{\{0,T\{} = \sum_t \gamma_t^{\pm} \, |\bar{Z}_n[t] - Z_n[t]| + \phi_t.$$

Adding Kuhn-Tucker multipliers μ_t allows to guaranty, using (5), that δ is an increasing function:

$$\delta = \text{argmin}_\delta d_{\{0,T\{} + \sum_{t, Z[t]=1, t'=\text{argmin}_{t'>t} \bar{Z}[t']=1} \mu_t \left[1 + \frac{\delta[t'] - \delta[t]}{t' - t}\right].$$

The minimization is to be initialized with $\delta[t] = 0$, i.e. considering only deletion/insertion and no shift.

A step further, in order to solve the ambiguity between indexing functions, from the last time to the previous time, we must choose a function $\gamma_t^{\pm}$ that decreases in the past, while its value must bounded by the higher shift cost, i.e. $\phi_t < \gamma_t^{\pm}$, for the maximal acceptable value of $\delta[t]$.

Mollification of the alignment metric. Let us finally proposes a mollification derived from the mollification proposed for the coincidence metric and taking into account the shifts defined by the alignment metric. Here we consider that the δ indexation is fixed, and propose as criterion:

$$\begin{array}{rcl} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{Z}\right) = \sum_{nt} {}_\upsilon d_{\{0,T\{}, & & \\ {}_\upsilon d_{\{0,T\{} & = & \gamma_t^{\pm} \, H_\upsilon\left(\left(1 - 2\,\bar{Z}_n[t + \delta[t]]\right)(V_n[t] - \theta)\right) \\ & + & \phi_t \left(Z_n[t] + (1 - Z_n[t])\, H_\upsilon\left(-\frac{\delta[t]}{\tau}\, H_\upsilon(\theta - V_n[t])_{\nu=0}\right)\right). \end{array}$$

Obviously, the term related to $\gamma_t^{\pm}$ enjoys the same properties as for the coincidence metric and we simply re-use what has been developed in this case.

The term related to ϕ_t has a different behavior depending on $Z_n[t]$]:
- If $Z_n[t] = 1$, i.e., when there are both a shift source and a shift target at time t, there is no adjustment on $V_n[t]$ since reducing one shift delay may increase the other one, with an unpredictable effect.
- If $Z_n[t] = 0$, i.e., when there is only a shift source at this time t, then the value of $V_n[t]$ is increased/decreased depending on the sign of $\delta[t]$ in order to reduce the absolute value of the delay:

$\delta[t] > 0$	$t < \bar{t}$	better shift t in the future	$V_n[t] \downarrow$
$\delta[t] < 0$	$\bar{t} < t$	better shift t in the past	$V_n[t] \uparrow$

in accordance with the signs of variation in the criterion. Furthermore, the variation of $V_n[t]$ is bounded in order the adjustment not to generate an additional

[2]Interesting enough is the fact we can apparently easily define a continuous form of this variational definition writing:

$$d\left(\bar{\mathbf{z}}, \mathbf{z}\right) = \min_\delta \int_t \gamma(t)^{\pm} \left[|\bar{z}_n(t + \delta(t)) - z_n(t)| + \phi_t\left(\frac{\delta(t)}{\tau}\right)\right] + \mu(t)(1 + \delta'(t))$$

with $\delta(t) = 0$ as initial value, writing $z(t) = \sum_{t_k^n} D(t - t_k^n)$, D being the Dirac distribution, and also define the related mollification, as proposed here in the discrete case.

spurious event. This is the reason why we consider $H_\upsilon(\theta - V_n[t])_{\nu=0}$ in order to maintain $\theta > V_n[t]$.

This last part if the specification is clearly a heuristic that has been adjusted when experimenting at the numerical level, in order to observe the proper behavior.

Conclusion. We have made explicit the different elements allowing us to propose (2) as a reasonable criterion. Given δ, the weights are adjusted in order to optimize the adequacy between the desired and expected event trains. At the implementation level, a relaxation minimization scheme is proposed:

1. Given a network event dynamic $\mathbf{Z}$ calculates δ to estimate the alignment distance;
2. For this indexing δ optimize the weights to tune $V_n[t]$ in order to reduce the alignment distance;
3. As soon as the event train is changed, repeat step 1 in order to re-estimate δ, stopping when a local minimum is found.

C Mollification of the Heaviside function

Let us specify how to mollify the event generation function $\rho() = H(u-\theta)$. The Heaviside function is defined here with $H(0) = 0$. In other words, we focus on the fact an event is defined by a state value above a given threshold θ. The generalization to other semi-algebraic conditions is straightforward, since they always can be stated as a combination of Heaviside functions (see [26] for a treatise on the subject).

In words, we precisely need to replace $H(u)$ by a regular function that can influence the estimation,
- either if the condition is incorrect,
- or if the condition is correct, but close to be incorrect, i.e. at margin ν of the correctness boundary, while
- we better require the function to have no influence if the condition is correct and beyond this boundary.

A suitable function that fits with this requirement is the $H_\upsilon(u)$ non-linear regular profile described in Fig. 3, and is defined as:

$$H_\upsilon(u) \stackrel{\text{def}}{=} H(u + \sqrt{\upsilon}\,\nu)\, \exp\left(-\frac{\upsilon}{u + \sqrt{\upsilon}\,\nu}\right), \tag{6}$$

thus with $H_\upsilon(u) = 0, u \leq -\sqrt{\upsilon}\,\nu$ and $H_\upsilon(+\infty) = 1$. Here ν is a *margin*, i.e. it allows one to maintain the state at a non-infinitesimal distance to the threshold.

It is easy to verify that this function is regular including in 0.

It is also important to notice that the function is convex on $]-\infty, \frac{\upsilon}{2} - \sqrt{\upsilon}\,\nu[$, i.e., when $H_\upsilon(u) < e^{-2}$ (drawn in *magenta* in Fig. 3).

Furthermore, $\lim_{\upsilon \to 0} H_\upsilon(u) = H(u)$, assuming that $H(0) = 0$. This convergence is uniform in $\mathcal{R} - [-\sqrt{\upsilon}\,\nu, -\sqrt{\upsilon}\,\nu + a[, a > 0$, thus in $[0, +\infty[$. This comes from the fact that $|H(u) - H_\upsilon(u)| < (1 - e^{-\sqrt{\upsilon}})$, thus uniformly bounded when $\upsilon \to 0$. This would not have been the case without a margin $\nu > 0$. More precisely, the reader can verify, that this is the denominator with an additive term $\upsilon^d\,\nu, d < 1$, say $d = 1/2$, which guaranties the convergence and the uniformity of this convergence in $[0, +\infty[$. Around $-\sqrt{\upsilon}\,\nu$, however, it is not possible to have convergence uniformity, which seems to be intrinsically related to the discontinuity of H. As a consquence, at the numerical level values must not remain in this neighborhood, to avoid "spurious jumps" when $\upsilon \to 0$.

A step further, $|H - H_\upsilon|_2$ is bounded for the L^2 norm (which is not the case for the L^1 norm), since $\int_{-\infty}^{+\infty} (H(u) - H_\upsilon(u))^2 du \Big|_{\upsilon=1,\nu=0} = 2\,\log(2)$, so that $|H - H_\upsilon|_2$ is finite, which is also an important fact for numerical algorithms.

Since $H_\upsilon(u) = 0, u \leq -\sqrt{\upsilon}\,\nu$, while still regular, the influence of obviously correct events is avoided, allowing the criterion to better tune more critical events.

More precisely :

$$H'_\upsilon(u) = H_\upsilon(u)\ \frac{\upsilon}{(u + \sqrt{\upsilon}\,\nu)^2} = H\left(u + \sqrt{\upsilon}\,\nu\right)\ \frac{\upsilon}{(u + \sqrt{\upsilon}\,\nu)^2} + O\left(\frac{1}{u}\right),$$

while $H_\upsilon(u) < H(u)$ for $u > 0$.

The derivation is not obvious since formally:

$$H'_\upsilon(u) = b(u + \sqrt{\upsilon}\,\nu) + H_\upsilon(u)\ \tfrac{\upsilon}{(u+\sqrt{\upsilon}\,\nu)^2}$$

with $b(u) \stackrel{\text{def}}{=} D(u)\,\exp\left(-\frac{\upsilon}{u}\right)$, $D()$ being the Dirac distribution. However, for all compact domain bounded function $\phi()$, we obtain:

$$\int b(u)\,\phi(u) = \left[\exp\left(-\tfrac{\upsilon}{u}\right)\ \phi(u)\right]_{u=0} = 0$$

thus $b() = 0$ in the distribution sense.

A step further, there is a closed-form for the n-th derivative, i.e.

$$\begin{array}{rcl} H_\upsilon^{(n)}(u) & = & H_\upsilon(u)\ \sum_{k=0}^{n-1} (-1)^k\,\mu_{k,n}\,\frac{\upsilon^{n-k}}{(u+\sqrt{\upsilon}\,\nu)^{2\,n-k}} \\ & = & H_\upsilon(u)\ \frac{\upsilon^n}{(u+\sqrt{\upsilon}\,\nu)^{2\,n}} + O\left(\frac{1}{u^{2\,n-1}}\right) \end{array},$$

for positive integers, iteratively defined as:

$$\mu_{k,n} = \begin{cases} 1 & k = 0 \\ \mu_{k,n-1} + (2\,n - k + 1)\,\mu_{k-1,n-1} & 0 < k < n \\ 0 & n \leq k \end{cases},$$

straightforward to derive from a piece of computer algebra.

Morally, this mollification corresponds to a convolution $H_\upsilon = \upsilon * H$ with respect to some suitable convolution kernel (without, up to our best knowledge, a closed form expression).

All together, the choice of the mollification kernel is not trivial. Better choices do exists, but they have to enjoy at least the properties made explicit here.

D Numerical verification

In order to estimate the variational method performance in terms of precision, we consider the simplest possible linear feed-forward network:

$$V_n[t] = \sum_m W_{nm}\, I_m[t] + \beta_n[t], Z_n[t] = H(V_n[t] - \theta)$$

where the the input $I_m[t]$ is a known deterministic signal and the additive noise $\beta_n[t]$ is drawn from a zero-mean random Gaussian distribution of standard deviation β. Using this model, we perform a master-slave validation test. The weights of the master are drawn from a Gaussian distribution of standard deviation σ and the additive noise of magnitude β, is added to the master output before presented to the estimation. The estimation is done on a slave noiseless linear feed-forward network of the same number of output units N and input units M, considering C trials of T time steps, which initial weights are similarly randomly drawn. Here we use $N = 4$, $M = 2$, $T = 1000$, $C = 2$ with $\theta = 0.5$. And typically consider $\sigma = 1$, $\beta = 0.25$ unless specified. Default alignment metric parameters are $\upsilon = 1$, $\mu = 0.1$ and $\tau = 0$ unless their values are explored.

Typical results are illustrated in Fig. 4, where it is numerically shown that the estimation using the proposed alignment variational estimation has the same performance as the "ideal" least-square estimation on the state values for small data errors. This is not a trivial result since the alignment variational estimation mechanism can not observed the master state values but only the related events. Furthermore, we approximate here a non-regular discontinuous criterion by the mollification mechanism and there were no guaranty that it should work. Here it does for small and reasonable errors (up to 30% of noisy events in this particular case). Fig. 4 also shows that the dispersion for this numerical test is small, yielding a robust result.

A step further, we numerically verify in Fig. 5 that the mollification parameters values have only a marginal influence on the final estimation, in this case. The use of a margin yields better results, as hypothesized in the specification, while the mollification introduces a small additional bias reduced when the mollified approximation converges towards the true alignment metric, these effects being of the seconder order. We also verify in Fig. 5 that these results are stable for different values of time-shift cost.

E Discussion.

Let us propose a short discussion about general links between the present development and what is known regarding the computational power of artificial or biologically plausible neural networks with event-based (i.e., spiking) activity [1, 27, 28, 2].

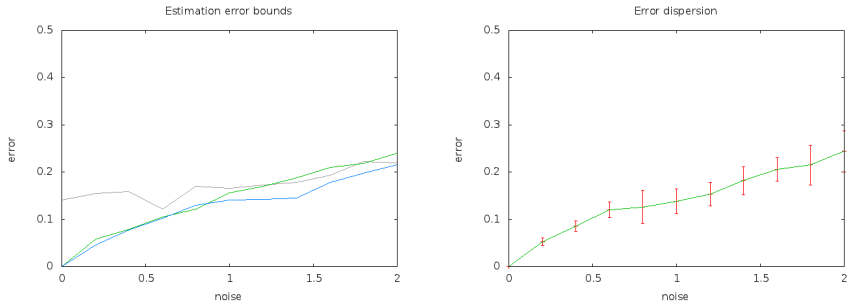


Fig. 4: *Estimation error bounds* : Alignment error obtained without any estimation (in gray), with a least-square estimation on the state values (in red), and using the proposed alignment variational estimation (in green), for different additive noise β in abscissa. The ordinate corresponds to the average alignment error form 0 for an exact estimation to 0.5 for a totally random result. The alignment mollification estimation is bounded by the non-estimation case from above an ideal least-square estimation where all state values (and not only the events) could be known from below. *Error dispersion* : Representation of the numerical test dispersion for the alignment metric estimation. Error-bars correspond to $\pm$ one standard-deviation.

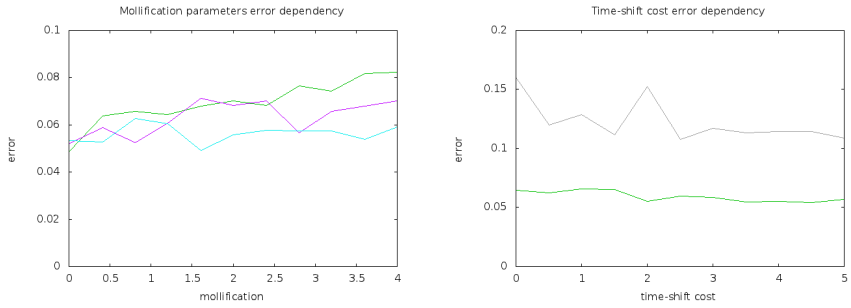


Fig. 5: *Mollification parameters error dependency* : Marginal influence on the final estimation of the mollification parameters v in abscissa, for three margin parameter values $\mu = \{0, 0.1, 0.2\}$ in $\{green, magenta, cyan\}$ respectively. The ordinate corresponds to the average alignment error. *Time-shift cost error dependency* : Comparison between the alignment error obtained without any estimation (in gray) with the proposed alignment variational estimation (in green), using the same conventions as for Fig. 4, showing the stability of the estimation not only for an event-count metric.

The key problem of calculability: non-learnability.

It is known that recurrent neuron networks with frequency rates are universal approximators [29], as multilayer feed-forward networks are [30]. This means that neuron networks are able to simulate dynamical systems, not only to approximate measurable functions on a compact domain, as originally stated (see, e.g., [29] for a detailed introduction on these notions).

Spiking neuron networks have been proved to be also universal approximators [27] and that, theoretically, spiking neurons can perform very powerful computations with precise event timings. Spiking neurons are at least as computationally powerful as the sigmoidal neurons traditionally used in artificial neuron networks [31, 32]. This result has been shown using a spike-response model (see [33] for a review) and considering piece-wise linear approximations of the membrane potential profiles. In this context, analog inputs and outputs are encoded by temporal latencies of event firings. It has been shown that any feed-forward (multi-layer) or recurrent analog neuronal network (e.g. Hopfield network) can be simulated arbitrarily closely by an insignificantly larger network of spiking neurons. The assertion holds even in the presence of noise [31, 32]. Such theoretical results highly motivate the use of spiking neuron networks for modelling and simulation purpose.

The key point, however, is the non-learnability of spiking neurons [12], since it is proved that this problem is NP-complete, when considering the estimation of both weights and delays. Here we show that we can "elude" this caveat and propose an alternate efficient estimation, inspired by biological models.

We also have to notice, that the same restriction apply not only to simulation but, as far as this model is biologically plausible, also holds at the biological level. It is thus an issue to wonder if, in biological neuron networks, delays are really estimated during learning processes, or if a weaker form of weight adaptation, as developed here, is considered.

As far as this contribution is concerned, we consider a weak notion of biological plausibility: A simulation is biologically plausible if it verifies an explicit set of constraints observed in biology. More precisely, we have taken time constraints, shared by all dynamics, further called "general time constraints". The time constraints are based on biological temporal limits and appear to be very precious quantitative elements, both for estimating the coding capacity of a system and for improving simulations.

Considering such learnability constraints, i.e., how can artificial or biological systems by-pass such computational barrier ?

By-passing the non-learnability barrier.

As pointed out previously, the weights estimation is proved to be NP-complete. This means that in order to "learn" the proper parameters we have to "try all possible combinations of delays". This is intuitively due to the fact that each delay has no "smooth" effect on the dynamics but may change the whole dynamics in an unpredictable way.

This is the way proposed to elude this NP-complete problem by considering *another* estimation problem. Here we do not estimate *one* delay (for each synapse) but consider connection weights at several delays and then estimate a balancing of their relative contribution. This means that we consider a *weak* delay estimation problem.

The alternative approach is to estimate delayed weights, i.e. a quantitative weight value W''_{ojd} or W'_{ijd} at each delay $d \in \{1, D\}$. Obviously, the case where there is a weight W_{ij} with a corresponding delay $d_{ij} \in \{0, D\}$ is a particular case of considering several delayed weights, since we can write $W_{ijd} = W_{ij}\,\delta(d - d_{ij})$, $\delta()$ being the Kronecker symbol in this case. In other words, with our weaker model, we are still able to estimate a neuron network with adjustable synaptic delays.

We thus do not restrain the neuron network model by changing the problem, but enlarge it. In fact, the present estimation provides a smooth approximation of the previous NP-complete problem.

The key idea of finding an approximate solution to the previous NP-complete problem, is instantiated at the implementation level, using mollified metrics, and the interest of such approach has been stated considering numerical examples.

The main limit of the present approach: polychronization.

A spiking network can polychronize, i.e., exhibit reproducible time-locked but not synchronous firing patterns within 1 millisecond precision. Polychronization can be viewd as a generalization of the notions of synchronization and synfire chains. Due to the interplay between the delays and a form synaptic plasticity, the spiking neurons spontaneously self-organize into groups and generate patterns of stereotypical polychronous activity.

In [34], it has been shown that the number of co-existing polychronous groups far exceeds the number of neurons in the network, resulting in an unprecedented memory capacity of the system. The author speculates on the significance of polychrony to the theory of neuronal group selection and cognitive neural computations.

In [35], the network processing and the resulting performance is explained by the concept of polychronization, The model emphasizes that polychronization can be used as a tool for exploiting the computational power of synaptic delays and for monitoring the topology and activity of a spiking neuron network.

Taking such complex aspects of the neural code into account cannot be performed by any available metrics. New metrics, taking long term interactions into account have to be developed and this is a challenging issue. This is not the case here and a challenging perspective of the present development.