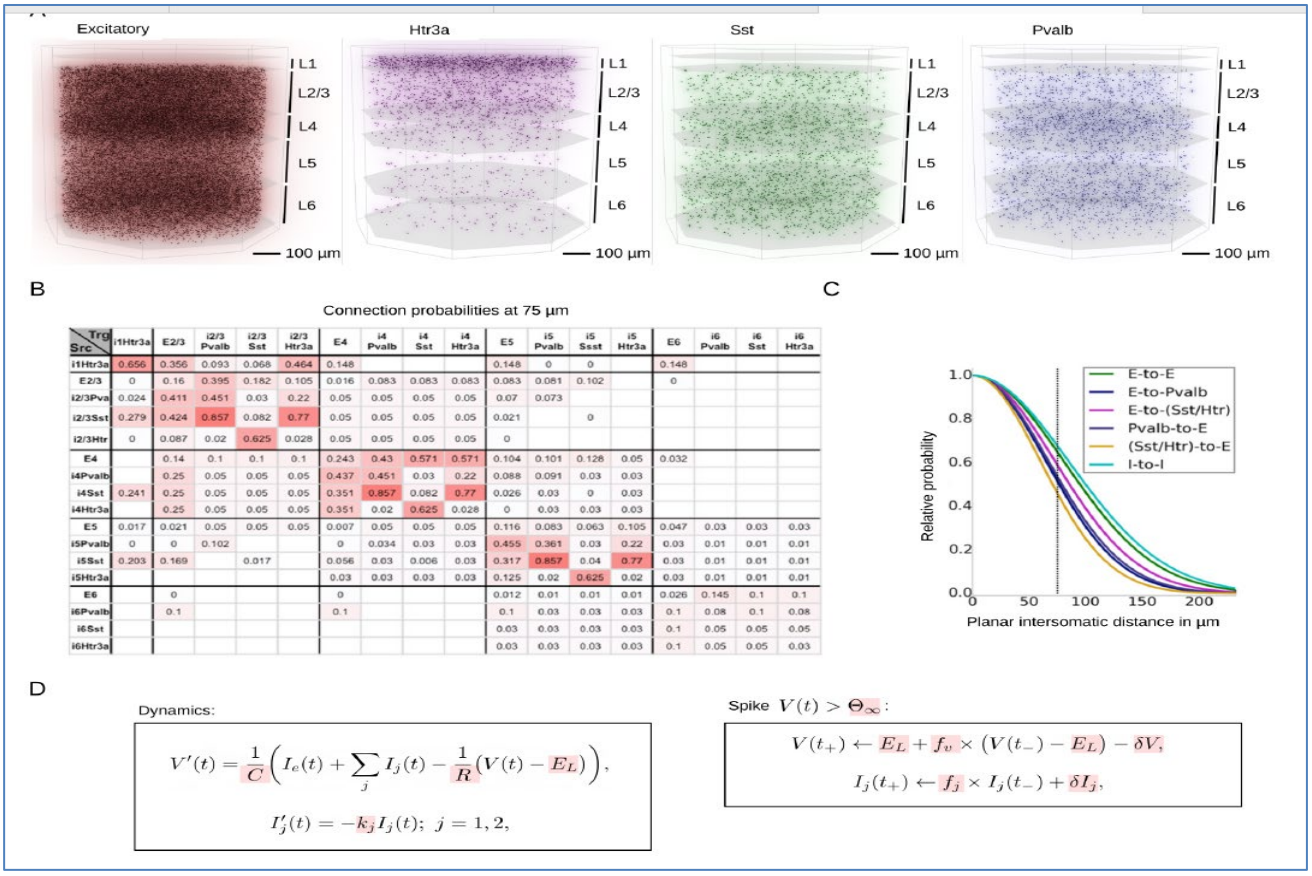
Framework for deep learning scheme for spike-based applications (D3.10 - SGA3)



Trg / Src	i1Htr3a	E2/3	i2/3 Pvalb	i2/3 Sst	i2/3 Htr3a	E4	i4 Pvalb	i4 Sst	i4 Htr3a	E5	i5 Pvalb	i5 Ssst	i5 Htr3a	E6	i6 Pvalb	i6 Sst	i6 Htr3a
i1Htr3a	0.656	0.356	0.093	0.068	0.464	0.148				0.148	0	0		0.148			
E2/3	0	0.16	0.395	0.182	0.105	0.016	0.083	0.083	0.083	0.083	0.081	0.102		0			
i2/3Pvalb	0.024	0.411	0.451	0.03	0.22	0.05	0.05	0.05	0.05	0.07	0.073						
i2/3Sst	0.279	0.424	0.657	0.082	0.77	0.05	0.05	0.05	0.05	0.021		0					
i2/3Htr3a	0	0.087	0.02	0.625	0.028	0.05	0.05	0.05	0.05	0							
E4		0.14	0.1	0.1	0.1	0.243	0.43	0.571	0.571	0.104	0.101	0.128	0.05	0.032			
i4Pvalb		0.25	0.05	0.05	0.05	0.437	0.451	0.03	0.22	0.088	0.091	0.03	0.03				
i4Sst	0.241	0.25	0.05	0.05	0.05	0.351	0.857	0.082	0.77	0.026	0.03	0	0.03				
i4Htr3a		0.25	0.05	0.05	0.05	0.351	0.02	0.625	0.028	0	0.03	0.03	0.03				
E5	0.017	0.021	0.05	0.05	0.05	0.007	0.05	0.05	0.05	0.116	0.083	0.063	0.105	0.047	0.03	0.03	0.03
i5Pvalb	0	0	0.102			0	0.034	0.03	0.03	0.455	0.361	0.03	0.22	0.03	0.01	0.01	0.01
i5Sst	0.203	0.169		0.017		0.056	0.03	0.006	0.03	0.317	0.857	0.04	0.77	0.03	0.01	0.01	0.01
i5Htr3a						0.03	0.03	0.03	0.03	0.125	0.02	0.625	0.02	0.03	0.01	0.01	0.01
E6		0				0				0.012	0.01	0.01	0.01	0.026	0.145	0.1	0.1
i6Pvalb		0.1				0.1				0.1	0.03	0.03	0.03	0.1	0.08	0.1	0.08
i6Sst										0.03	0.03	0.03	0.03	0.1	0.05	0.05	0.05
i6Htr3a										0.03	0.03	0.03	0.03	0.1	0.05	0.05	0.03

Figure 1: Large-scale cortical model trained with BPTT, tested on a 52k-neuron patch

Project Number:	945539	Project Title:	HBP SGA3

Document Title:	Framework for deep learning scheme for spike-based applications
Document Filename:	D3.10 (D29) SGA3 M42 RESUBMITTED 231208.docx
Deliverable Number:	SGA3 D3.10 (D29)
Deliverable Type:	Other
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA3 M42 / 30 SEP 2023
Actual Delivery Date:	SGA3 M42 / 19 SEP 2023 (resubmitted 08 Dec 2023)
Author(s):	Wolfgang MAASS, TUGRAZ (P55)
Compiled by:	Wolfgang MAASS, TUGRAZ (P55)
Contributor(s):	Sacha VAN ALBADA, JUELICH (P20), contributed to Section 2.1 Thomas NOWOTNY, UoS (P106), contributed to Section 2.2 Guozhang CHEN, TUGRAZ (P55), contributed to Section 2.3 Mihai PETROVICI, UBERN (P71), contributed to Section 2.4
WP QC Review:	Anita VAN OERS, UM (P117)
WP Leader / Deputy Leader Sign Off:	Rainer GOEBEL, UM (P117)
T7.4 QC Review:	Annemieke MICHELS. EBRAINS (P1)
Description in GA:	Finalized and tested version of algorithms for advanced learning, implemented in software made available to a general audience (not limited to HBP, addressing the EBRAINS user community at large).
Abstract:	A new suite of algorithms and software that can be used by a wide variety of users in computational and cognitive neuroscience for training brain models to exhibit desired computational functions, and by users from neuromorphic engineering and neurorobotics to port deep learning to their spike-based applications.
Keywords:	Spiking Neural Networks, Large-Scale Brain Modelling, Neural Network Optimisation
Target Users/Readers:	neuromorphic engineers, computational neuroscience community, computer scientists, funders, general public, HPC community, neuroscientific community, neuroscientists, platform users, researchers, scientific community, students

Table of Contents

Table of Figures

History of Changes made to this Deliverable (post Submission)

Date	Change Requested / Change Made / Other Action
19 SEP 2023	Deliverable submitted to EC
	Resubmission with specified changes requested in Review Report Main changes requested: The deliverable is acceptable in terms of scientific content, but the quality of the text is not sufficient and needs to be revised before public release. Either add details from or clearly list links to peer-reviewed publications or preprints related to the reported outcomes.
	Revised draft sent by WP to PCO. Main changes made, with indication where each change was made: • Change 1 More details added on implementations of learning rules • Change 2 Links added to references • Change 3 Details added about performance of learning rule implementations (see Figures 2-3)
	Revised version resubmitted to EC by PCO via SyGMa

1. Introduction

The work detailed in this Deliverable delves deep into the realm of computational neuroscience, aiming to bridge the gap between theoretical brain models and practical applications in neurorobotics and neuromorphic engineering. Our primary objective is to develop and optimise algorithms that can train brain models to exhibit desired computational functions, thereby paving the way for advanced spike-based applications. This endeavour aligns seamlessly with the overarching goals of the Human Brain Project (HBP) – to understand the human brain and harness its capabilities for societal benefits.

Goal/Aim of the Research: The primary aim of our research is to develop a suite of algorithms and software that can train detailed large-scale brain models, enabling them to perform desired computational functions. This has vast implications for neuromorphic engineering and neurorobotics, potentially revolutionising the way we approach deep learning with spike-based applications.

Major Scientific Hurdle: We will demonstrate the HBP's capability to overcome the challenge of integrating complex neural algorithms with practical applications, ensuring scalability, efficiency, and real-world relevance.

HBP Scientific Area Contribution: Our work significantly contributes to the 'Modelling' and 'Network Complexity' scientific areas within the HBP. It also touches upon aspects of 'Brain States' and 'Cognitive Functions', given the nature of the algorithms and their potential applications.

Interested Communities: Neuroscientists, neuromorphic engineers, AI and machine learning experts, neurorobotics professionals, and tech industry stakeholders would find our work particularly intriguing. The advancements we're making have the potential to reshape multiple domains, from academic research to commercial applications.

Downstream Applications: The outcomes of our research have vast downstream applications. In the realm of AI, our algorithms can lead to more efficient and scalable neural networks. In medicine, understanding brain models can aid in neurological disorder research. Additionally, our work has implications for advanced robotics, prostheses, and even ethical considerations in neurotechnology.

In essence, our research is pushing the boundaries of what is possible in computational neuroscience, opening doors to a myriad of applications that can transform our world.

2. Computational Neuroscience Algorithms and Software Suite

2.1 E-Prop Algorithm Implementation by Jülich Team

Eligibility propagation (e-prop) is a learning algorithm that uses neural eligibility traces to train spiking neural networks using only locally available information [1]. The algorithm was originally implemented using TensorFlow, but enabling scaling it to large networks and combining it with biologically based modelling approaches benefits from implementation in NEST. To this end, the Jülich team ported the e-prop algorithm to NEST, tested and optimised the performance of the implementation iteratively in different scaling scenarios, and studied the impact of different approximations; see the preliminary reports in poster abstracts [https://juser.fz-juelich.de/record/1007044; https://juser.fz-juelich.de/record/1006851]. They demonstrated that the implemented plasticity model performs well in large, sparsely connected networks, including the microcircuit model by Potjans and Diesmann [2]. Figure 2 shows the performance of the NEST implementation of e-prop on an example pattern generation task and its comparison with the original TensorFlow implementation.

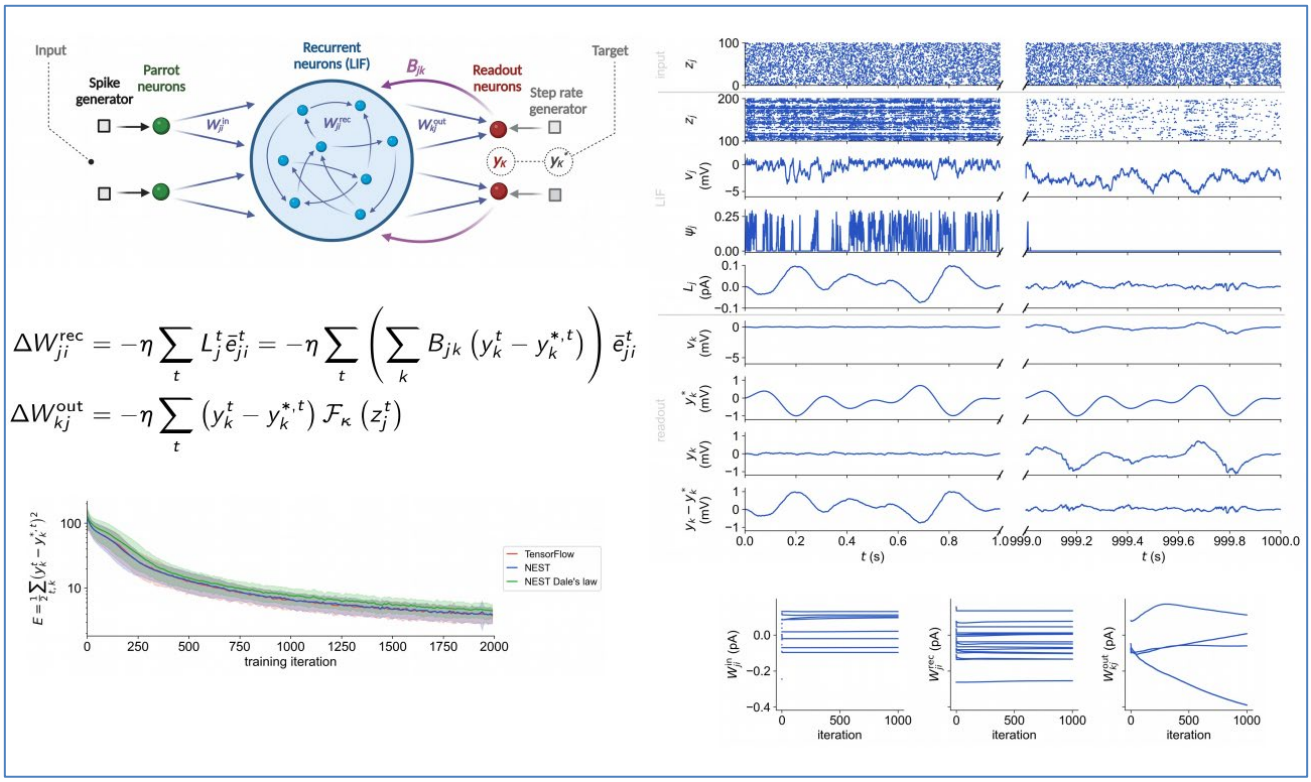


Figure 2: Performance of the NEST implementation of e-prop on a temporal pattern generation task

Top left: Setup of the network. Equations: Learning rules for the recurrent and output weights. η: learning rate, $\bar{e}_{ji}^t$: eligibility trace, z_j^t: spiking activity, y_k^t: network readout, $y_j^{*,t}$: target output, B_{jk}: feedback weights, L_j^t: learning signal, $\mathcal{F}_k$: filter applied to the spiking activity. Bottom left: Evolution of the error across training iterations. Right: Evolution of the model variables over time during training.

1) P1998 - Bellec, G., Scherr, F., Subramoney, A. *et al.* A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat Commun* **11**, 3625 (2020). https://doi.org/10.1038/s41467-020-17236-y

2) Tobias C. Potjans, Markus Diesmann, The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Model, *Cerebral Cortex*, Volume 24, Issue 3, March 2014, Pages 785–806, https://doi.org/10.1093/cercor/bhs358

2.2 E-Prop and EventProp Implementations by UoS

UoS has (as part of Task T5.10, and a HBP Partnering Project) implemented the e-prop algorithm for Long short-term memory Spiking Neural Networks (LSNNs) and the EventProp algorithm for recurrent spiking neural networks in the GeNN software. This software can be used by researchers in Computational Neuroscience and Machine Learning researchers to train spiking neural networks to exhibit desired behaviours.

UoS has demonstrated that the e-prop implementation in GeNN is comparable in training time to PyTorch-based BPTT algorithms, but scales to much larger network sizes [1]. In addition, LSNNs implemented in GeNN have an order of magnitude faster inference than BPTT trained networks in TensorFlow: LSNNs with one layer of 256 hidden units trained on the sequential MNIST dataset using e-prop had 10 times lower inference latency than LSTMs running on the same hardware and a lower Energy Delay Product (EDP) across all tested batch sizes (1-1024). Additionally, UoS compared this model against an implementation on the Loihi neuromorphic hardware and found that, although LSNN inference on Loihi requires less energy, GeNN models run faster [1].

The GeNN EventProp implementation in turn is an order of magnitude faster in training than e-prop, see Figure 3 and [2] for the Eventprop work.

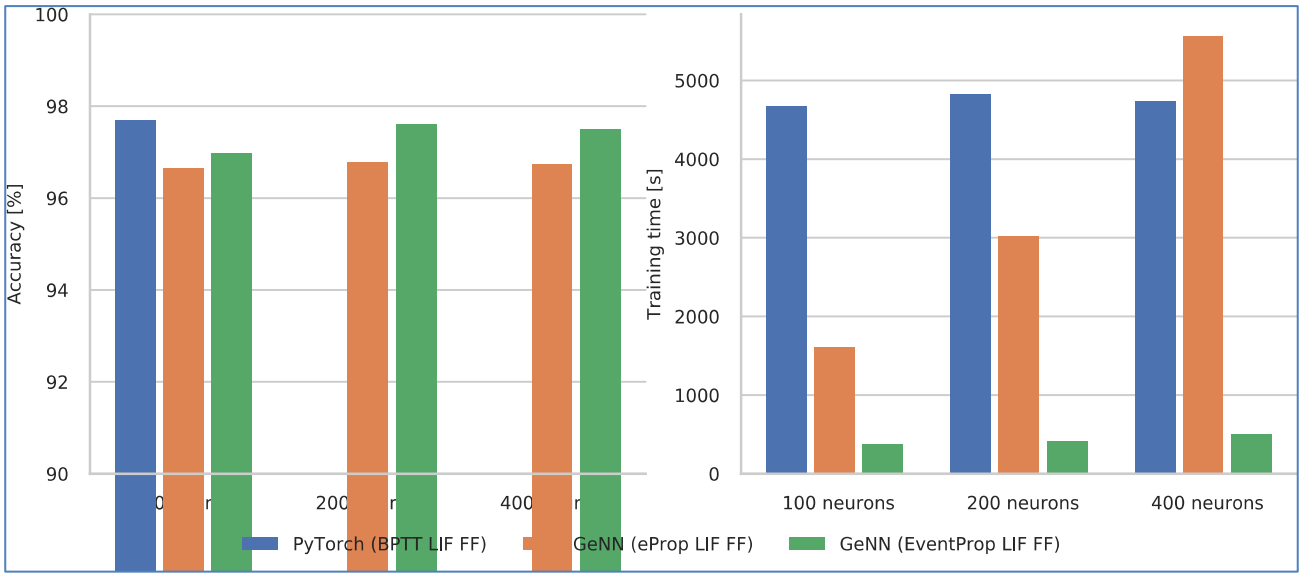


Figure 3: Accuracy and computational performance of eprop and EventProp implementations

Feedforward networks were used with one hidden layer of size N= 100, 200, 400 neurons were used to classify the latency encoded MNIST dataset. Both e-prop and EventProp achieve comparable, albeit slightly lower accuracy than when training with back-propagation through time (BPTT) in PyTorch. However, for small networks e-prop in GeNN was significantly faster but lost this advantage as the network was scaled up. EventProp in GeNN was more than 10x faster than BPTT in PyTorch for all tested network sizes.

Finally, UoS has shown how e-prop can be combined with structural plasticity to achieve sparsely connected networks with even better computational performance while suffering minimal losses in task accuracy: Larger classifiers with 1 or 2 hidden layers were trained using e-prop on the DVS-gesture database, using combinations of feedforward and recurrent layers of different sizes. These models provide state-of-the-art accuracy and trained faster than models trained with comparable approaches such as FPTT. Furthermore, GeNN can exploit sparse connectivity to further accelerate the training, see [3] for full details

The code is available on GitHub at https://github.com/neworderofjamie/eprop_paper, https://github.com/neworderofjamie/ml_genn_eprop_paper and https://github.com/tnowotny/genn_eventprop. The software is offered in the new mlGeNN framework developed at UoS, which is continuing on separate project funding, see https://github.com/genn-team/ml_genn.

1) P3773 - Knight, James C, and Thomas Nowotny. 2022. “Efficient GPU Training of LSNNs Using eProp.” In Neuro-Inspired Computational Elements Conference, New York, NY, USA: ACM, 8–10. DOI: 10.1145/3517343.3517346

2) P4026 - Nowotny, T., Turner, J. P., & Knight, J. C. (2022). Loss shaping enhances exact gradient learning with EventProp in Spiking Neural Networks. arXiv preprint arXiv: 2212.01232. DOI: 10.48550/arXiv.2212.01232

3) P4028 – Knight, James C, and Thomas Nowotny. 2023. “Easy and Efficient Spike-Based Machine Learning with mlGeNN.” In Neuro-Inspired Computational Elements Conference, San Antonio TX USA: ACM, 115–20. DOI: 10.1145/3584954.3585001

2.3 Version of BPTT that is applicable to detailed models of cortical microcircuits, by TU Graz

TUGraz has developed and extensively tested an approximation to BPTT (back-propagation through time) that can be applied to detailed large-scale models for parts of the neocortex. This method complements the methods described in sections 2.1 and 2.2. It is optimized for achieving both the best computational performance of cortical microcircuit models after training, and for running rapidly on a single GPU. In fact, simulation and training of the large-scale model for a generic cortical microcircuit consisting of 51,978 neurons of 111 different types (modelling a patch with 800 mikron diameter) of V1 from (Billeh et al., 2020), see Figure 1, requires just 8 times more time than biological time, i.e., the time which the circuit needs to carry out the computation in vivo (without any training, just producing its dynamics). This is apparently the first time that such a large-scale data-based cortical microcircuit model could be trained to carry out demanding computational tasks. We trained it simultaneously to carry out 5 different computational tasks. Four of them have commonly been used for behavioural training of mice, and numerous experimental data are available on the neural activity in vivo during task performance.

One remarkable new insight is that the data-based model can be trained with fewer examples than simpler control models of the same size, see Figure 4 of [1]. Another remarkable insight is that the large-scale data-based model does not overfit, although one usually finds this when large NN models are trained for tasks of this type. In fact, the data-based model turns out to be remarkably robust to noise in the network and in sensory input that had not been present during training.

The training algorithm and the results of training are described in two publications [1] and [2]. Reference [1] contains all details of the training method. Reference [2] shows in addition that one can reverse-engineer the organization of computations in the cortical microcircuit that are installed through this training method, and that one can identify causal relationships between specific details of the neurophysiological data and their impact on the network computation.

1) P4035 - Chen, G., Scherr, F., & Maass, W. (2022). A data-based large-scale model for primary visual cortex enables brain-like robust and versatile visual processing. Science Advances, 8(44), eabq7592. 10.1126/sciadv.abq7592

2) P4033 - Chen, G., Scherr, F., & Maass, W. (2023). Data-based large-scale models provide a window into the organization of cortical computations. bioRxiv, 2023-04. 10.1101/2023.04.28.538662

2.4 Deep learning frameworks for biological and bio-inspired networks, by UBERN

Based on the core idea of error correction, both globally (at the network level) and locally (at the level of individual neurons and synapses), UBERN has developed a series of models for deep learning in physical neuronal systems. The emphasis is on physical computing in the brain or neuromorphic hardware, which require signals to be available locally, where and when they are needed

(spatiotemporal locality), and processing happens by the natural dynamics of the substrate. This is in opposition to logical computing in Turing machines or von-Neumann-type architectures, where the relevant mathematical calculations are carried out by a finite-state machine or arithmetic-logical unit that is specifically designed for such numerical manipulations, and where data is constantly ferried between compute and memory (the von Neumann bottleneck).

Network dynamics, and in particular synaptic plasticity, are derived from high-level, normative functions that describe the state or performance of the system as a whole. These can be an energy [1], a Lagrangian [2], a measure of statistical similarity [3] or a deviation in spike timing [4]. The resulting dynamics relate directly to existing biological observations and explain several features of cortical networks that were previously not directly linked to behaviour or function. Furthermore, they point towards new aspects of neuronal structure and dynamics that can be tested in future experiments. Finally, the spike-based models allow a fast and efficient implementation on neuromorphic hardware, which rivals the state of the art in combined speed and power consumption.

For a more detailed description, see also Deliverable D3.11, as well as the publications below:

1) P2949 - Haider, P., Ellenberger, B., Kriener, L., Jordan, J., Senn, W., & Petrovici, M. A. (2021). Latent equilibrium: A unified learning theory for arbitrarily fast computation with arbitrarily slow neurons. Advances in Neural Information Processing Systems, 34, 17839-17851. https://papers.nips.cc/paper/2021/file/94cdbdb84e8e1de8a725fa2ed61498a4-Paper.pdf

2) P3937 - Senn, W., Dold, D., Kungl, A. F., Ellenberger, B., Jordan, J., Bengio, Y., Sacramento, J. & Petrovici, M. A. (2023). A neuronal least-action principle for real-time learning in cortical circuits. eLife, 12. 10.7554/elife.89674.1

3) P2621 - Kreutzer, E., Senn, W., & Petrovici, M. A. (2022). Natural-gradient learning for spiking neurons. eLife, 11. 10.7554/elife.66526

4) P2917 - Göltz, J., Kriener, L., Baumbach, A., Billaudelle, S., Breitwieser, O., Cramer, B., Dold, D., Kungl, A., Senn, W., Schemmel, J., Meier, K. & Petrovici, M. A. (2021). Fast and energy-efficient neuromorphic deep learning with first-spike times. Nature machine intelligence, 3(9), 823-835. 10.1038/s42256-021-00388-x

2.5 Looking Forward

Preliminary tests have shown that the described methods can also be applied to substantially larger neural network models. Furthermore, we found that the learning speed could be improved by combining these methods with Learning-to-Learn techniques.