What should a neuron aim for? Designing local objective functions based on information theory

Andreas C. Schneider[1,2,*] **Valentin Neuhaus**[1,2,*] **David A. Ehrlich**[3,*] **Abdullah Makkeh**[3]
Alexander S. Ecker[4,1] **Viola Priesemann**[1,2,†] **Michael Wibral**[3,†]

[1]Max Planck Institute for Dynamics and Self-Organization, Göttingen
[2]Institute of Physics, University of Göttingen
[3]Campus Institute for Dynamics of Biological Networks, University of Göttingen
[4]Institute of Computer Science and Campus Institute Data Science, University of Göttingen
andreas.schneider@ds.mpg.de, valentin.neuhaus@ds.mpg.de
davidalexander.ehrlich@uni-goettingen.de

Abstract

In modern deep neural networks, the learning dynamics of the individual neurons is often obscure, as the networks are trained via global optimization. Conversely, biological systems build on self-organized, local learning, achieving robustness and efficiency with limited global information. We here show how to enhance the interpretability of the individual artificial neurons' function by developing a local learning framework similar to that of biological neurons. The local objective function is parameterized using a recent extension of information theory – Partial Information Decomposition (PID) – which decomposes the information that a set of information sources holds about an outcome into unique, redundant and synergistic contributions. Our framework enables neurons to locally shape the integration of information from various input classes by selecting which of the inputs should contribute uniquely, redundantly or synergistically to the output. This selection is expressed as a learning goal for an individual neuron, which can be directly derived from intuitive reasoning or via numerical optimization, offering a window into task-relevant local information processing. Achieving performance on par with backpropagation while preserving neuron-level interpretability, our work advances a principled information-theoretic foundation for local learning strategies.

1 Introduction

Most artificial neural networks (ANNs) optimize a single global objective function through algorithms like backpropagation, orchestrating parameters across all computational elements of the network as a unified computational structure. While this global objective approach has proven effective across a large variety of tasks, the role of the individual neuron often remains elusive, hindering the understanding of how local computation contributes to global task performance. In contrast, biological neural networks exhibit a markedly different approach to learning, relying strongly on self-organization between neurons and locally available information only [1]. Interestingly, these limitations come with the advantage that their computation can also be interpreted locally, at least in principle. Combining this local interpretability with the observation that biological neural networks achieve high levels of efficiency and robustness even in the absence of a centrally coordinated objective raises the question: Could ANNs benefit from similar local learning mechanisms to enhance the *local* interpretability of their computations, while maintaining performance?

[*]These authors contributed equally to this work and share first authorship.
[†]These authors contributed equally to this work and share last authorship.

1^{st} Workshop on NeuroAI @ 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Neuroscience models have indeed shown that local learning rules [2–4] can solve a variety of tasks, such as sequence learning, unsupervised clustering, and classification tasks [2, 5, 6]. They are formulated on the basis of temporal coincidence (spike-timing) and thus do not offer direct insights into the actual information processing at each neuron. To facilitate such functional insight, more recently *information theory* has been used to formulate more abstract, but interpretable frameworks for local learning rules (e.g., [7, 8]). Information theory [9] allows one to abstractly prescribe how much of the information from different inputs should be conveyed to the output of a neuron [7]. Moreover, it provides a general and flexible mathematical framework, which captures the information processing at an abstract and interpretable level free of details of the exact implementation. As such, information theory has been employed to define general optimization goals, including global objectives like cross-entropy loss [10] and more localized ones such as local greedy infomax [11], and early attempts at passing only information that is coherent between multiple inputs [12].

Nevertheless, classical information theory falls short of describing how multiple sources work together to produce an output. In particular, it lacks the capacity to describe how the information from different sources is integrated in redundant, synergistic or unique ways in the output. These different ways, or information *atoms*, however, contribute very differently to a neuron's function: Redundant information represents coherent information from multiple sources, unique information reflects what a single source adds, and synergy arises from considering sources together. This complexity can be quantified by using the recent framework of Partial Information Decomposition (PID) [13–15]. Through the decomposition of the total information into information atoms, one can paint a comprehensive, yet interpretable picture of how the different input variables contribute to the local computation of an individual neuron. PID has already been utilized to analyze the information representation and flows in DNNs [16, 17]. We turn around that approach, and employ PID to directly formulate goal functions on the individual neuron level. These "infomorphic neurons" then have the ability to optimize for encoding specific parts of the information they receive from their inputs, allowing for an application to tasks from supervised, unsupervised and memory learning, as demonstrated explicitly in [18].

The main contributions of this work are as follows: (1) We use information theory and PID to formulate interpretable, per-neuron goal functions for neurons with three input classes ("trivariate"), overcoming the limitations of previous approaches with only two classes ("bivariate"), (2) We systematically optimize the goal function parameters for these "infomorphic" neurons, provide an intuitive interpretation, and thereby we provide insights into the local computational goals of typical classification tasks, and (3) For classification tasks we show that PID-based local learning can achieve performance comparable to backpropagation, while being interpretable on a per-neuron basis.

2 Infomorphic neurons

Neurons can be viewed as information processors that receive a number of input signals and process them to produce their own output signal. The output Y of a neuron can be considered a random variable, and the total output information can be quantified using the Shannon entropy $H(Y)$. This output information consists of two parts: the mutual information $I(Y : \boldsymbol{X})$ coming from the inputs $\boldsymbol{X}$ and the residual entropy $H(Y|\boldsymbol{X})$ arising from stochastic processes within the neuron.

Mutual information can be used to quantify the amount of information that is carried by different input classes about the neuron's output. Inspired by the dendritic compartments of pyramidal neurons, we consider the aggregated feedforward (F), contextual (C) and lateral inputs (L) as sources $\boldsymbol{X}$ individually (see Fig. 1). Using PID, the total mutual information $I(Y : F, C, L)$ between the output of the neuron Y and the three aggregated inputs F, C and L is dissected into 18 PID *atoms* with intuitive interpretations (see Fig. 1.**C** and Table 2): For instance, the *unique* information of the feedforward connection about the output Y, $\Pi_{\{F\}}$, is the information which can only be obtained from the feedforward, but neither from the contextual or lateral inputs, with the terms $\Pi_{\{C\}}$ and $\Pi_{\{L\}}$ being defined analogously. The *redundant* information $\Pi_{\{\mathrm{F}\}\{\mathrm{C}\}\{\mathrm{L}\}}$ is the information which can be equivalently obtained from either feedforward, contextual or lateral inputs about Y, while the *synergistic* information $\Pi_{\{\mathrm{F,C,L}\}}$ can only be obtained from all three inputs considered jointly. In general, the atoms Π_α describe redundancies between synergies and can be addressed by their *antichains* α, which are sets of sets of variables, with the inner sets describing synergies and the outer set redundancies between them [13].

However, these atoms remain underdetermined since there are 18 unknown atoms and only seven classical mutual information terms $I(Y : \boldsymbol{X})$ with $\boldsymbol{X} \subseteq \{F, C, L\}$ providing constraints through the so-called *consistency equations* [13]. For this reason, an additional quantity needs to be defined, which is usually a measure for redundancy [13, 15]. Such a definition for redundancy, based on the concept of shared exclusions in probability space, is given by the shared-exclusion PID, $I_\cap^{\mathrm{sx}}$ [15] (see Appendix A.1).

Depending on the task the network is set to solve, different PID atoms become relevant to the information processing. We focus here on the application to supervised learning tasks, in which the ground-truth label is provided as the context C during training. The intuitive goal for the neuron is to maximize the redundant information between F and C which is unique with respect to L, given by the atom $\Pi_{\{\mathrm{F}\}\{\mathrm{C}\}}$, to capture only the feedforward signal that agrees with the label and to avoid encoding the same information in multiple neurons.

Quantitatively, such general PID goals can be formulated as an objective function comprising a linear combination of PID atoms as $G = \sum_\alpha \gamma_\alpha \Pi_\alpha + \gamma_{\mathrm{res}} H_{\mathrm{res}} = \boldsymbol{\gamma}^T \boldsymbol{\Pi}$, where the residual entropy $H_{\mathrm{res}} = H(Y|F, C, L)$ is included in the vector $\boldsymbol{\Pi}$ for brevity of notation. Due to the differentiability of the $I_\cap^{\mathrm{sx}}$ redundancy measure, G can be optimized by adjusting the weights of F, C and L using gradient ascent.

In a concrete implementation, an infomorphic neuron first aggregates its input into the three input signals F, C and L as weighted sums, which are then passed into an activation function to stochastically produce a firing signal. In the learning step, the aggregation weights are updated to improve the PID goal function. For more detail on the concrete implementation including pseudocode, refer to Appendix A.4.

3 Experiments and Results

Experiments We demonstrate how an infomorphic network with one hidden layer can be set up using either intuitive or optimized goal function parameters to solve the MNIST supervised handwritten image classification task [19]. We devised a network architecture consisting of an input layer, a hidden layer made up from N_{hid} trivariate infomorphic neurons and an output layer consisting of 10 bivariate infomorphic neurons. We explored three slightly different approaches to how the hidden layer can be set up using trivariate infomorphic neurons (see Appendix A.3), illustrated in Fig. 2.**A**. Each neuron has an activation function, which in principle can be chosen arbitrarily (For more details, see Appendix A.4), that takes the weighted aggregation of the inputs F, C and L to produce a value used as a probability to stochastically output "1" (firing) or "-1" (non-firing). The parameters $\boldsymbol{\gamma}$ for a PID goal function can be obtained in two distinct ways: Firstly, they can be derived from intuitive notions about the nature of the computations necessary to solve the problem, as explained before. Secondly, they can also be optimized using a suitable hyperparameter optimization procedure (see Appendix A.5). The learning of the goal function is explained in Appendix A.6.

Performance The performance of all major setups over hidden layer size is shown in Fig. 2.**B**. The three trivariate infomorphic setups use the same set of optimized goal parameters (see Appendix A.10) for all hidden layer sizes and significantly outperform the random baseline as well as the networks with a bivariate setup for the hidden layer (see appendix Fig. 8). Setup 1 matches the performance of backpropagation for up to 100 neurons, and reaches its maximum test accuracy for 500 neurons before its performance starts decreasing with larger layer sizes. This decrease in accuracy can be attributed to a lack of convergence of the neurons (appendix Fig. 4), likely arising through the interaction of too many neurons. To alleviate the convergence problem, we performed additional experiments (setup 2) with the number of lateral connections reduced to maximally 100, which leads to better convergence and contributes to the continuous increase of performance for larger hidden layers, reaching a median test accuracy of 97.5% for 2000 neurons, slightly below the 98.0% of the corresponding backpropagation benchmark. Finally, experiments with setup 3 provide evidence that the direct label input to the hidden layer can be replaced with feedback from the next layer, while still enabling solid performance especially for large layers.

Goal parameters The optimized goal used for training the main three setups at all sizes were optimized using setup 1 for $N_{\mathrm{hid}} = 100$ neurons on the validation set. More details regarding those optimizations and the optimized goal functions can be found in Appendix A.10. The optimized

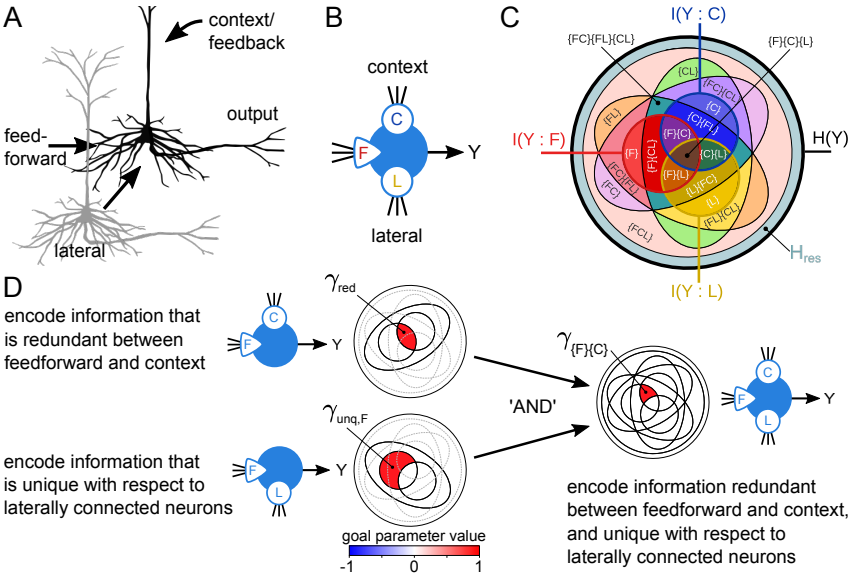

Figure 1: **Infomorphic Neurons directly optimize PID-based goals locally to solve a global task. A,B.** Infomorphic neuron with three inputs: feed-forward (F), contextual (C) and lateral (L) provide an information-theoretic abstraction of the different input classes found in biological neurons such as pyramidal neurons. **C.** With three input classes, 18 distinct PID atoms can be differentiated, represented by 18 different colors, plus the residual entropy H_{res} in the outer circle. Classical information-theoretic quantities such as the entropy $H(Y)$ and mutual information I with individual sources are depicted by ovals, indicating how they can be built from PID atoms. **D**. Disentangling three input classes allows one to optimize for complex local goals based on 19 distinct terms. Here, we show how trivariate PID allows to combine two bivariate objective functions (see Appendix A.2): In a supervised learning task, one might want to maximize information in the neuron's output that is redundant between the feedforward input F and label C, while simultaneously ensuring the neuron's output stays unique with respect to lateral neurons L. While bivariate goal functions would only allow for optimizing one of these objectives at a time, both objectives can be combined to the goal of maximising only the single atom $\Pi_{\{F\}\{C\}}$ in the trivariate case.

objective functions outperform the intuitive goal function by including additional PID atoms besides the intuitively derived $\Pi_{\{F\}\{C\}}$.

Interpretation Compared to the intuitive $G = \Pi_{\{F\}\{C\}}$, a better performing goal function $G \approx 0.33\Pi_{\{F\}\{C\}\{L\}} + \Pi_{\{F\}\{C\}} - \Pi_{\{F\}\{L\}} - \Pi_{\{FC\}\{FL\}}$ was found using the hyperparameter optimization approach. This optimized goal function is interpretable since we can interpret the individual atoms. For the MNIST task neurons aim for (i) encoding information that is redundant between feedforward and context inputs and thus task relevant, but not already encoded in other neurons (strongly positive $\gamma_{\{F\}\{C\}}$), (ii) avoid encoding information that is not in the context, thus not task-relevant, and already encoded by other neurons (strongly negative $\gamma_{\{F\}\{L\}}$) and (iii) allow redundancies of task relevant information between neurons (slightly positive $\gamma_{\{F\}\{C\}\{L\}}$). Additionally, neurons avoid synergies that require F and either C or L for their recovery ($\gamma_{\{FC\}\{FL\}}$) (see Table 3 for an overview of the parameter values and interpretations). For the CIFAR10 task (see Fig. 5.**D**), the two most important parameters $\gamma_{\{F\}\{C\}\{L\}}$ and $\gamma_{\{F\}\{C\}}$ point to a prioritization of redundancy over uniqueness between neurons.

4 Discussion

We introduce a novel local learning framework for ANNs using partial information decomposition, making neural learning objectives more interpretable. Our framework expresses this local objective by selecting which output information should be uniquely, redundantly or synergistically determined by the various classes of local inputs to a neuron. We derive a simple, intuitive learning objective for

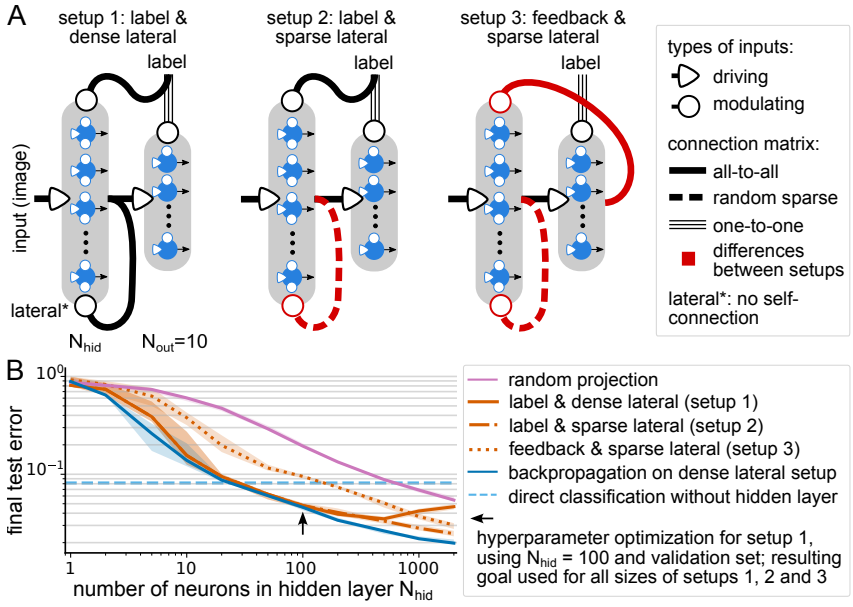

Figure 2: **Infomorphic networks with three input classes approach the performance of a similar network trained with backpropagation on the MNIST handwritten digit classification task.** **A**. In the hidden layer of the network, the infomorphic neurons receive feedforward and lateral connections as well as either the ground-truth label or feedback from the output layer, outlined in three setups (detailed in Appendix A.3). **B**. Infomorphic networks achieve similar performance as the same network trained with backpropagation. For larger layers, using a sparse connectivity significantly improves performance (setup 2). The context signal in the hidden layer can either be given by the label or a feedback connection from the output layer (setup 3). The lines indicate mean values, with the intervals depicting the maximum and minimum of 10 runs.

supervised learning and upon some optimization of this simple rule show that its performance on MNIST matches that of backpropagation-trained ANNs.

Related works We are unaware of other efforts to use PID for designing ANNs with local learning objectives, aside from the bivariate infomorphic networks in [18]. The most related work is Kay and Phillips' coherent infomax [20], which inspired our use of PID-based redundancy as a neural goal function. However, as analyzed in [8], the absence of PID at the time limited their approach's expressiveness and clarity for more than two input classes. Our work shows that three input classes—data to transform (F), relevance of the data (C), and reducing the redundancy with other available information (L)—are crucial for effective information-theoretic learning.

Beyond the formulation of information theoretic learning rules, there is literature using PID to analyze deep neural network function, such as [17, 16, 21–25]. These studies necessarily differ from the perspective on neural function taken here, as they analyze the function of neurons in feedforward networks trained with backprop, where the backprop signals are not directly available as inputs to the neural activation function. This is in contrast to our infomorphic networks where the training signals are just regular inputs to the neurons. Certainly also related to our work is a line of research on the information bottleneck principle [26]. The main idea is that a representation of input data should best encode as much information as possible about the task, while reducing the amount of information that is irrelevant to the task [27–29].

Limitations and Outlook We show that the infomorphic local optimization approach reaches a performance comparable to backpropagation in a setup with one fully connected hidden layer. These infomorphic neurons lay the foundation for promising new research directions such as analyzing deeper networks or neurons with continuous activations.

For the study of biological neural networks, infomorphic networks could potentially serve as a powerful constructive modeling approach. While information-theoretic gradients are likely too

complex for direct biological implementation, neurons may approximate similar objectives through simpler local learning rules. Having a principled and general way to formulate, identify and test local information processing objectives could provide a new perspective on local learning in general.

In conclusion, the formulation of local objective functions in the language of PID directly enables an interpretation of the information processing of neurons. Our work represents an important step towards a principled theory of local learning founded in information theory.

Code Availability The framework and the code to reproduce the results of this work are available under https://github.com/Priesemann-Group/Infomorphic_Networks.

Acknowledgments and Disclosure of Funding

We would like to thank Marcel Graetz, Jonas Dehning, Mark Blümel, Fabian Mikulasch and Lucas Rudelt for their input and fruitful discussions about this topic. We would also like to thank Johannes Zierenberg and Sebastian Mohr and the rest of the Priesemann and the Wibral Group for their valuable comments and feedback on this work. A.S., V.N. and V.P. were funded via the MBExC by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy-EXC 2067/1-390729940. A.S., V. N., V.P., and M.W., were supported and funded by the DFG via the RTG 2906 "Curiosity" project ID 502807174. D.E. and M.W. were supported by a funding from the Ministry for Science and Education of Lower Saxony and the Volkswagen Foundation through the "Niedersächsisches Vorab" under the program "Big Data in den Lebenswissenschaften" – project "Deep learning techniques for association studies of transcriptome and systems dynamics in tissue morphogenesis". A.M. and M.W. are employed at the Campus Institute for Dynamics of Biological Networks (CIDBN) funded by the Volkswagenstiftung. A.E., V.P. and M.W. received funding from the DFG via the SFB 1528 "Cognition of Interaction" - project-ID 454648639. A.E. was supported from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101041669). M.W. was supported by the flagship science initiative of the European Commission's Future and Emerging Technologies program under the Human Brain project, HBP-SP3.1-SGA1-T3.6.1.

References

[1] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

[2] Donald Hebb. The organization of behavior., 1949.

[3] Yang Dan and Mu-ming Poo. Spike timing-dependent plasticity of neural circuits. *Neuron*, 44 (1):23–30, 2004.

[4] Sen Song, Kenneth D Miller, and Larry F Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9):919–926, 2000.

[5] Peter Földiak. Forming sparse representations by local anti-hebbian learning. *Biological cybernetics*, 64(2):165–170, 1990.

[6] Benjamin Cramer, David Stöckel, Markus Kreft, Michael Wibral, Johannes Schemmel, Karlheinz Meier, and Viola Priesemann. Control of criticality and computation in spiking neuromorphic networks with plasticity. *Nature communications*, 11(1):2853, 2020.

[7] Jim Kay. *Neural networks for unsupervised learning based on information theory*, pages 25–63. Oxford University Press, Inc., United States, 2000.

[8] Michael Wibral, Viola Priesemann, Jim W Kay, Joseph T Lizier, and William A Phillips. Partial information decomposition as a unified approach to the specification of neural goal functions. *Brain and cognition*, 112:25–38, 2017.

[9] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[11] Sindy Löwe, Peter O'Connor, and Bastiaan Veeling. Putting an end to end-to-end: Gradient-isolated learning of representations. *Advances in neural information processing systems*, 32, 2019.

[12] Jim W Kay and WA Phillips. Coherent infomax as a computational goal for neural systems. *Bulletin of mathematical biology*, 73(2):344–372, 2011.

[13] Paul L Williams and Randall D Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

[14] Aaron J Gutknecht, Michael Wibral, and Abdullah Makkeh. Bits and pieces: Understanding information decomposition from part-whole relationships and formal logic. *Proceedings of the Royal Society A*, 477(2251):20210110, 2021.

[15] Abdullah Makkeh, Aaron J Gutknecht, and Michael Wibral. Introducing a differentiable measure of pointwise shared information. *Physical Review E*, 103(3):032149, 2021.

[16] David Alexander Ehrlich, Andreas Christian Schneider, Viola Priesemann, Michael Wibral, and Abdullah Makkeh. A measure of the complexity of neural representations based on partial information decomposition. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=R8TU3pfzFr`.

[17] Tycho Tax, Pedro AM Mediano, and Murray Shanahan. The partial information decomposition of generative neural network models. *Entropy*, 19(9):474, 2017.

[18] Abdullah Makkeh, Marcel Graetz, Andreas C Schneider, David A Ehrlich, Viola Priesemann, and Michael Wibral. A general framework for interpretable neural learning based on local information-theoretic goal functions. *arXiv e-prints*, pages arXiv–2306, 2023.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Jim Kay and William A Phillips. Activation functions, computational goals, and learning rules for local processors with contextual guidance. *Neural Computation*, 9(4):895–910, 1997.

[21] Seiya Tokui and Issei Sato. Disentanglement analysis with partial information decomposition. *arXiv preprint arXiv:2108.13753*, 2021.

[22] Shujian Yu, Kristoffer Wickstrøm, Robert Jenssen, and Jose C Principe. Understanding convolutional neural networks with information theory: An initial exploration. *IEEE transactions on neural networks and learning systems*, 32(1):435–442, 2020.

[23] Paul Pu Liang, Zihao Deng, Martin Ma, James Zou, Louis-Philippe Morency, and Ruslan Salakhutdinov. Factorized contrastive learning: Going beyond multi-view redundancy. *arXiv preprint arXiv:2306.05268*, 2023.

[24] Paul Pu Liang, Chun Kai Ling, Yun Cheng, Alex Obolenskiy, Yudong Liu, Rohan Pandey, Alex Wilf, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal learning without labeled multimodal data: Guarantees and applications. *arXiv preprint arXiv:2306.04539*, 2023.

[25] Salman Mohamadi, Gianfranco Doretto, and Donald A Adjeroh. More synergy, less redundancy: Exploiting joint mutual information for self-supervised learning. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1390–1394. IEEE, 2023.

[26] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[27] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[28] Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31): 7937–7942, 2018.

[29] Ziv Goldfeld and Yury Polyanskiy. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1):19–38, 2020.

[30] Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying unique information. *Entropy*, 16(4):2161–2183, 2014.

[31] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL `https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf`.

[32] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[34] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[36] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[37] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

A Appendix

A.1 Definition of shared-exclusion redundancy

In this section, we briefly motivate and explain the shared-exclusion redundancy measure $I_\cap^{\mathrm{sx}}$ introduced by Makkeh et al. [15]. The mutual information $I(T : S_1, S_2)$ between a target variable T and two source variables S_1 and S_2 can be interpreted in a Bayesian sense as an average measure for how the prior belief of the target event $T = t$, needs to be updated in light of the event of observing *both* source events $S_1 = s_1$ *and* $S_2 = s_2$ simultaneously:

$$I(T : S_1, S_2) = \sum_{t,s_1,s_2} \mathbb{P}(T = t, S_1 = s_1, S_2 = s_2) \log_2 \frac{\mathbb{P}(T = t | S_1 = s_1 \wedge S_2 = s_2)}{\mathbb{P}(T = t)}.$$

Makkeh et al. [15] build on this logic and define redundancy as an average measure for how the beliefs about the target event $T = t$ need to be updated if instead it is only known that $S_1 = s_1$ *or* $S_2 = s_2$ have occurred:

$$I_\cap^{\mathrm{sx}}(T : S_1, S_2) = \sum_{t,s_1,s_2} \mathbb{P}(T = t, S_1 = s_1, S_2 = s_2) \log_2 \frac{\mathbb{P}(T = t | S_1 = s_1 \vee S_2 = s_2)}{\mathbb{P}(T = t)}.$$

For more than two source variables s_i (where i is an index enumerating the set of source variables), the term to condition on becomes a disjunction between conjunctions of the form $\bigvee_{\boldsymbol{a}\in\alpha} \bigwedge_{i\in\boldsymbol{a}} S_i = s_i$ for the redundancy associated with the antichain α. The atoms Π can then be computed from these redundancies via a Moebius inversion, as laid out in detail in Gutknecht et al. [14].

The definition is symmetric with respect to permutation of the sources, fulfills a target chain rule and is differentiable with respect to the underlying probability distribution [15], which makes it a suitable definition for optimizing objective functions.

A.2 Bivariate infomorphic neurons

The infomorphic neurons introduced in [18] considered only the apical and basal dendrites of the pyramidal neurons as input classes lumping the lateral connection into any of these input classes. In these bivariate infomorphic neurons the total mutual information $I(Y : F, C)$ between the output of the neuron Y and two aggregated inputs F and C can be dissected into four PID *atoms* (see Fig. 3.**C**): The *unique* information of the feedforward connection about the output Y, $\Pi_{\mathrm{unq,F}}$, is the information which can only be obtained from the feedforward and not the context input, with the unique information $\Pi_{\mathrm{unq,C}}$ of the context being defined analogously. The *redundant* information Π_{red} reflects the information which can equivalently obtained from either feedforward or contextual inputs about Y, while finally the *synergistic* information Π_{syn} can only be obtained from both inputs considered jointly. All classical mutual information terms between the target and subsets of source variables can be constructed from these PID atoms through the consistency equations [13]

$$\begin{aligned} I(Y : F, C) &= \Pi_{\mathrm{red}} + \Pi_{\mathrm{unq,F}} + \Pi_{\mathrm{unq,C}} + \Pi_{\mathrm{syn}} \\ I(Y : F) &= \Pi_{\mathrm{red}} + \Pi_{\mathrm{unq,F}} \\ I(Y : C) &= \Pi_{\mathrm{red}} + \Pi_{\mathrm{unq,C}}. \end{aligned} \tag{1}$$

However, these atoms are underdetermined as there are four unknown atoms with only three consistency equations providing constraints [13]. For this reason, an additional quantity needs to be defined, which is usually a measure for redundancy [13, 15]. Such a definition for redundancy, based on the concept of shared exclusions in probability space, is given by the shared-exclusion PID, $I_\cap^{\mathrm{sx}}$ [15] (see Appendix A.1). Importantly, the resulting PID is differentiable with respect to the underlying probability mass function, which allows for a specification and subsequent optimization of learning goals based on PID.

Depending on the task the network is set to solve, different PID atoms become relevant to the information processing. In this work, we focus on the application to supervised learning tasks, in which the ground-truth label is provided as the context C during training. Here, the intuitive goal for the neuron is to foster redundancy between the feedforward and context inputs in its output, to capture only the feedforward signal that agrees with the label. Likewise, if the goal is unsupervised encoding of the input, lateral connections between neurons might be used as the context signal and

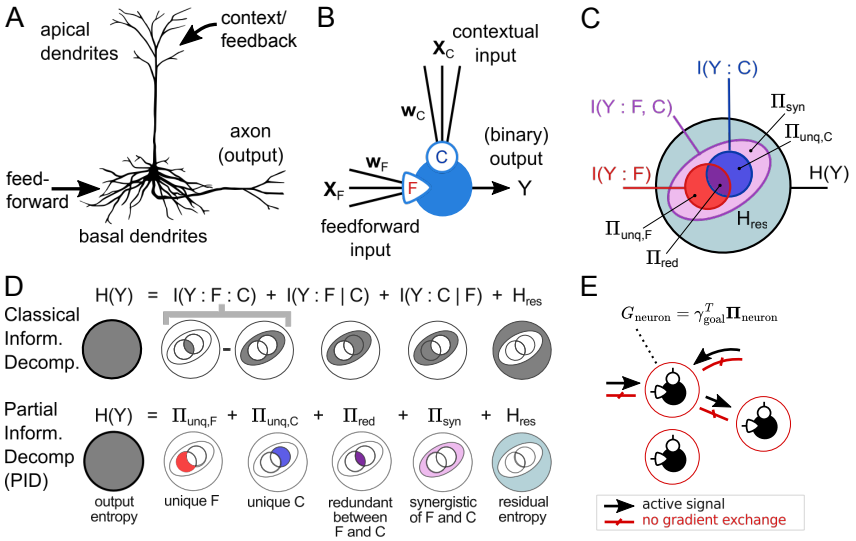

Figure 3: **Infomorphic neurons are abstract, information-theoretic neurons inspired by the structure of pyramidal neurons [8]. They are trained by adjusting their synaptic weights according to a PID-based goal function. A,B**. Inspired by the distinction between apical and basal dendrites in cortical pyramidal neurons, infomorphic neurons are defined as computational units with separate feedforward (F) and contextual (C) input classes. **C**: Partial information decomposition (PID) allows one to dissect the total entropy of the neuron into explainable components. **D**. PID enables one to distinguish which information comes *uniquely* from either the feedforward F ($\Pi_{\mathrm{unq,F}}$) or contextual C input ($\Pi_{\mathrm{unq,C}}$), whether it could be retrieved *redundantly* from the two (Π_{red}), or whether the two inputs contribute *synergistically* (Π_{syn}). Classic information theory (top) cannot disentangle these information atoms: The classic entities cover several of the atoms, and fundamentally lack one constraint, so that effectively one can only measure redundant minus synergistic information. Therefore, novel approaches to quantify PID have been developed in the past years [13, 14, 30]. **E**: Formulating goal functions G_{neuron} in terms of PID-atoms $\mathbf{\Pi}_{\mathrm{neuron}}$ enables one to formulate how strongly redundant, unique, or synergistic information should contribute to a neuron's output. Figure adapted from [18].

the goal might become maximizing the unique information of the feedforward input about the output, i.e., to capture only the information which is not already encoded in other neurons [18].

PID can not only be used to describe the local information processing, but also to optimize it through the maximization of a PID based objective function. Generally, such an objective function can be expressed as a linear combination of PID atoms as

$$G = \gamma_{\mathrm{red}}\Pi_{\mathrm{red}} + \gamma_{\mathrm{unq,F}}\Pi_{\mathrm{unq,F}} + \gamma_{\mathrm{unq,C}}\Pi_{\mathrm{unq,C}} + \gamma_{\mathrm{syn}}\Pi_{\mathrm{syn}} + \gamma_{\mathrm{res}}H_{\mathrm{res}} = \boldsymbol{\gamma}^T\mathbf{\Pi}, \tag{2}$$

where the residual entropy $H_{\mathrm{res}} = H(Y|F,C)$, although not being a PID atom, is included in the vector $\mathbf{\Pi}$ for brevity of notation. Due to the differentiability of the $I_\cap^{\mathrm{sx}}$ redundancy measure, the goal function can be optimized by adjusting the weights of the incoming connections of the inputs F and C using gradient ascent – with the analytic formulation of the gradients given in [18], or using a suitable autograd algorithm.

While such infomorphic neurons can indeed be used for supervised learning tasks [18], two input classes are insufficient for fully self-organizing the necessary local information processing in supervised tasks. This is because if in supervised learning each neuron is tasked with encoding information that is redundant between the whole label and the feedforward signal with no regard to what the other neurons already encode, the neurons will all inevitably start encoding much of the same information, leading to poor performance. To avoid this, the neurons need to be made aware of what the other neurons encode, similar to the unsupervised example. This can be achieved by incorporating a third input class representing lateral connections between neurons of the same layer, as presented next.

A.3 Network Setup

The trivariate neurons in the hidden layer each receive the whole image as their feedforward signal and the output of other neurons of the same layer through lateral connections. They additionally receive one of two choices for the context signal: Either the ground-truth label, which is only provided during training, or a feedback connection from the readout layer. The bivariate neurons in the output layer receive all outputs of the hidden layer and additionally exactly one element of the one-hot label as context, which is only provided during training, and are trained to maximize redundancy between their input and the information about the one label they observe.

The reasoning behind the three different approaches to how the hidden layer can be set up using trivariate infomorphic neurons (see Fig. 2.**A**) is as follows. In setup 1, we use the fully connected feedforward F as input, fully connected label as C and the output of all neurons of the same layer as all-to-all connected lateral input L. In setup 2, a sparse connectivity of a maximum of 100 connections is used instead of all-to-all for L. In setup 3, additionally to sparse L, the label is replaced by a fully connected feedback from the output layer as context C, indicating a path towards how multiple hidden layers may be stacked in the future. For comparison, we trained a benchmark network with the same connectivity as setup 1 using standard backpropagation and cross-entropy loss. Additionally, we trained the output layer of a fixed-random hidden layer setup using step-function activation.

A.4 Neuron structure and activation function

To construct infomorphic networks, a number of practical decisions need to be made for how the output signal is constructed from the inputs in the forward pass. As a first step, the higher-dimensional signals from the different input classes are aggregated to the single numbers F, C and L by a weighted sum. Subsequently, these aggregated inputs are passed into an activation function, which can be chosen arbitrarily, but needs to fit the requirements of the specific application. For the supervised classification task at hand, the function $A(F, C, L) = F[(1-\alpha_1-\alpha_2)+\alpha_1\,\sigma(\beta_1 FC)+\alpha_2\,\sigma(\beta_2 FL)]$ has been chosen, where σ refers to the sigmoid function and $\alpha_1 = \alpha_2 = 0.1$ and $\beta_1 = \beta_2 = 2$ are parameters that shape the influence of the input compartments on the activation function. Note that this function makes a clear distinction between the driving feedforward input and the modulatory context and lateral inputs, ensuring that the network performs similarly during training, when context and lateral inputs are provided, and for evaluation, where the context signal is withheld. The output of the activation function is finally mapped to the unit interval by a sigmoid function, whose output is used as a probability to stochastically output "1" (firing) or "-1" (non-firing). For the bivariate output layer, the sigmoid is not sampled from but just interpreted as a firing probability, with the neuron with the highest value being used as the network prediction. Because lateral connections depend on the activity of the neurons in a previous timestep, the same input is presented twice.

Input: data, num_epochs
Output: trained model
INITIALIZE model;
foreach *epoch in range(num_epochs)* **do**
  foreach *(batch_samples, batch_labels) in data* **do**
    INITIALIZE last_outputs to zero;
    last_outputs ← forward(f=batch_samples, c=batch_labels, l=last_outputs);
    last_outputs ← forward(f=batch_samples, c=batch_labels, l=last_outputs);
    foreach *neuron in model* **do**
      TrainNeuron(y=last_outputs[neuron], f=batch_samples, c=batch_labels,
       l=last_outputs[other neurons]);
    end
  end
end
return *trained model*

Algorithm 1: TrainModel

Input: y, f, c, l
BIN continuous values f, c, l to 20 equally sized bins;
COUNT occurrences of tuples (y, f, c, l) in batch;
COMPUTE empirical probability masses p(y, f, c, l);
COMPUTE isx_redundancies from p(y, f, c, l);
COMPUTE pid_atoms from isx_redundancies;
goal ← scalar_product(goal_params, pid_quantities);
PERFORM autograd to maximize goal;
UPDATE neuron weights;

Algorithm 2: TrainNeuron

A.5 Hyperparameter optimization of goal parameters

For the hyperparameter optimization, we compared two established techniques: Firstly, we used the tree-structured Parzen Estimator (TPE) [31], that splits the sample points into two groups with high and low performance, fits a model to each group and draws the next sample points according to the ratio of probabilities. Secondly, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [32] was employed, which generates new evaluation points by sampling from a multivariate Gaussian distribution that was fitted to the best samples from previous iteration steps.

A.6 Discretization and local gradients

To optimize a PID-based goal function during training, the first step is to estimate the PID atoms. Since the original $I_\cap^{\mathrm{sx}}$ measure only works with discrete variables, the aggregated inputs from a batch of samples are first discretized to 20 levels each. Subsequently, the PID atoms are computed by means of a plug-in estimation and the current value of the goal function is determined. For the local gradient computation, the gradient of the individual neuron's objective function is backpropagated via the output to produce the neuron's weight updates.

A.7 Compute resources

For the figures in this paper, we performed a total of seven hyperparameter optimizations: Three TPE optimizations and three CMA-ES optimizations of MNIST objective function parameter and a single CMA-ES optimization for the CIFAR task. Each of these optimizations took ≤ 12 hours to compute on a HPC cluster node with 32 cpu cores and two NVIDIA A100 GPUs with $40\,\mathrm{GB}$ of VRAM each.

Including evaluation runs and earlier computations not shown in the results, we estimate to have utilized a total of 200 hours of compute time on a compute node equivalent to the one described earlier.

A.8 Implementation details and model parameters

We implemented infomorphic networks as a flexible and efficient python package using pytorch [33] for automatic differentiation of the local goal functions. Furthermore, the optuna [34] package was used to compute the TPE and CMA-ES hyperparameter optimizations as well as the computation of the parameter importance via the mean decrease in impurity.

In this paper, we use the MNIST [35] and CIFAR10 [36] datasets. The MNIST dataset consists of 70,000 grayscale images of handwritten digits, each sized 28x28 pixels. The dataset is split into a training set of 60,000 samples and a test set of 10,000 samples. The CIFAR10 dataset consists of 60,000 images with three color channels and a resolution of 32x32 pixels. Of the 60,000 images, 50,000 are in the training set and 10,000 are in the test set. For each of our training runs with either dataset, 20% of the training set samples are withheld randomly to be used for validation.

The framework and the code to reproduce the results of this work are available under https://github.com/Priesemann-Group/Infomorphic_Networks.

The parameters used to train the models shown in Fig. 2 are listed in Table 1. The backpropagation model was trained to optimize the cross-entropy loss between the prediction and the true label.

Table 1: Model and training parameters for the models shown in Fig. 2

Parameter	Backprop Model	IM Hidden Layer	IM Output Layer
N_{Epochs}	100	100	100
N_{Batch}	1024	1024	1024
Optimizer	Adam	Adam	Adam
Learning rate η	0.001	0.002	0.003
Weight decay	0.0	0.00035	0.00015
Number of bins per dim.	-	20	20
Binning ranges	-	(-20,20)	adaptive
Objective function	cross-entropy	trivariate γ	$\gamma = (-0.2, 0.1, 1.0, 0.1, 0.0)^T$

A.9 Model Dynamics During Training

As a measure for the dynamics of the neurons during training, we introduce the self-cosine distance of the feedforward receptive field as

$$D_c^{(t)} = 1 - \frac{\boldsymbol{w}_F^{(t-1)} \cdot \boldsymbol{w}_F^{(t)}}{\|\boldsymbol{w}_F^{(t-1)}\| \|\boldsymbol{w}_F^{(t)}\|} \tag{3}$$

where $\boldsymbol{w}_F$ corresponds to feedforward weights of a single neuron of the hidden layer. The median value of $D_c^{(t)}$ for the hidden neurons of a trivariate model are shown in Fig. 4. While the self-cosine distance consistently increases with layer size in the dense setups, it does not increase in the sparse setup after a layer size of 100 neurons which is also the number of lateral connections in all of the larger sparse layers.

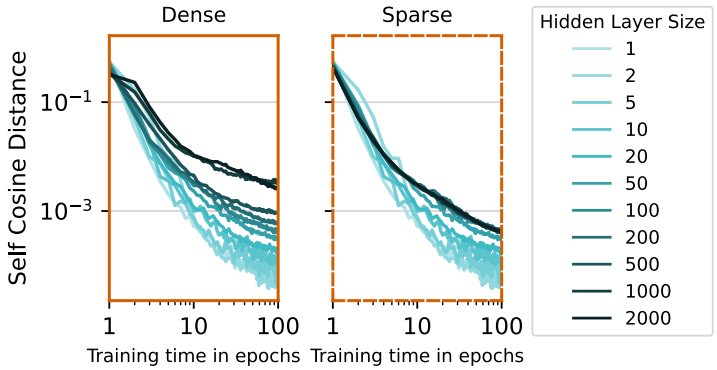

Figure 4: The median self-cosine distance of trivariate neurons for different layer sizes during the course of training in a dense (left) and a sparse (right) lateral connected setup.

A.10 Optimized Objective Functions

Optimization The optimization goal used for training the main three setups at all sizes were optimized using setup 1 for $N_{\text{hid}} = 100$ neurons on the validation set. In total, we performed six hyperparameter optimizations of 1000 trials, three using TPE and three using CMA-ES samplers, respectively. While the objective functions that were optimized with the TPE sampler included all γ parameters, the optimization with the CMA-ES sampler was only performed for the parameters that describe PID-atoms while the parameter for the residual entropy was set to 0. This was done because we observed a strong correlation between the parameter for the unique information of the feedforward input and the residual entropy which both had to be small. The goal function that performed best is illustrated in 5.**A** while an overview over all goal functions is shown in Fig. 6. Additionally to the optimizations performed on the MNIST dataset, we also performed a single CMA-ES optimization on the more complex CIFAR10 classification task [36], with the optimal parameters presented in Fig. 5.**D**.

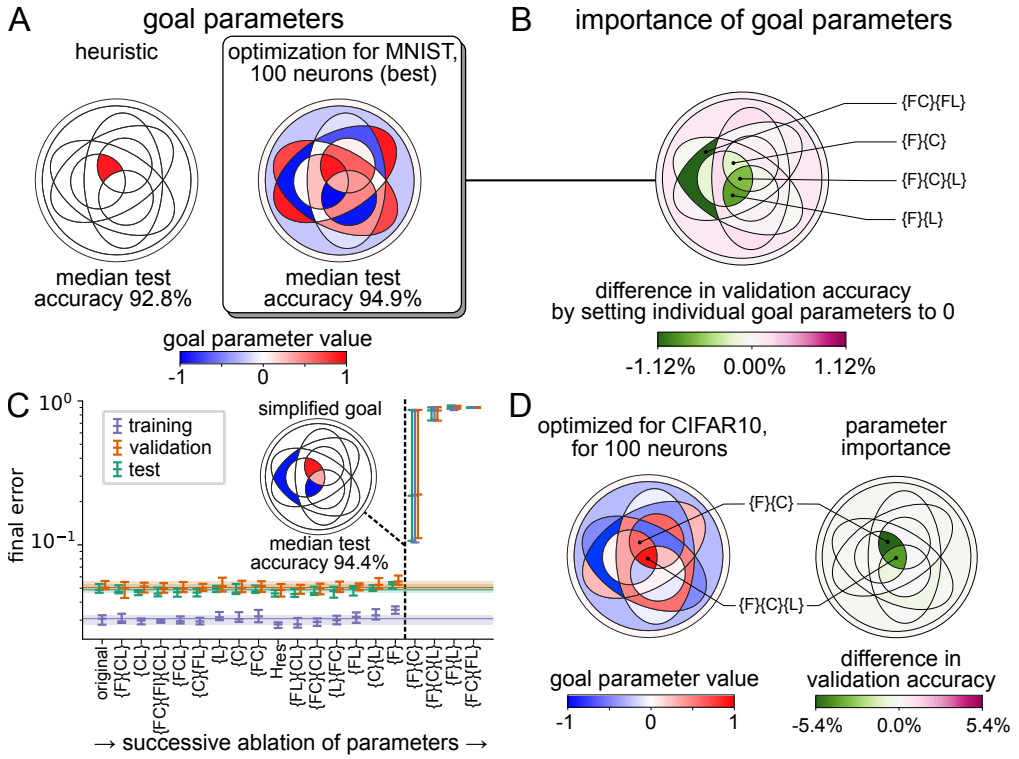

Figure 5: **Given a set of optimized goal parameters, an ablation study allows for the identification and interpretation of the most critical neural subgoals for a given task. A.** The heuristically defined goal function (see Fig. 1.**D**) shows 92.8% test accuracy in MNIST for $N_{hid} = 100$; optimizing the goal function using hyperparameter optimization increases the test accuracy to 94.9%. The optimized goal parameters include the heuristically found $\gamma_{\{F\}\{C\}}$, but also additional PID atoms to be maximized or minimized at the same time. **B.** For identifying the most important goal parameters, we performed an ablation study (individually setting parameters to 0) and measured the effect on network performance. **C.** A successive ablation of parameters in order from lowest to highest individual effect identifies four parameters as being crucial for network performance (see Table 3 for their definition and interpretation). The lines indicate mean values, with the intervals depicting the maximum and minimum of 10 runs. **D**. To test whether more complex image classification tasks require different atoms to be maximized, we perform a separate hyperparameter optimization for setup 1, 100 neurons in CIFAR10 and reach a median test accuracy of 42.5% (compared to 42.2% using backprop and 41.1% using the goal function originally optimized for MNIST).

Parameter importance An analysis of the optimized hyperparameter values (Fig. 5.**B**) shows that only relatively few parameters are of high importance for performance. By setting goal parameters to zero individually (Figure 4B, right) we find that there are four goal parameters that are critically non-zero for high performance on MNIST. This finding is confirmed by cumulatively setting goal function parameters to zero in the order of the individual drop in performance, which is illustrated in Fig. 5.**C**. Interestingly, we find that CIFAR10 two of the four critical goal parameters for MNIST seem especially important. As a complementary measure of parameter importance, the results of mean decrease in impurity [37] can be found in Fig. 6 for the six different MNIST optimizations.

A.11 The Atoms of the Partial Information Decomposition for Three Source Variables

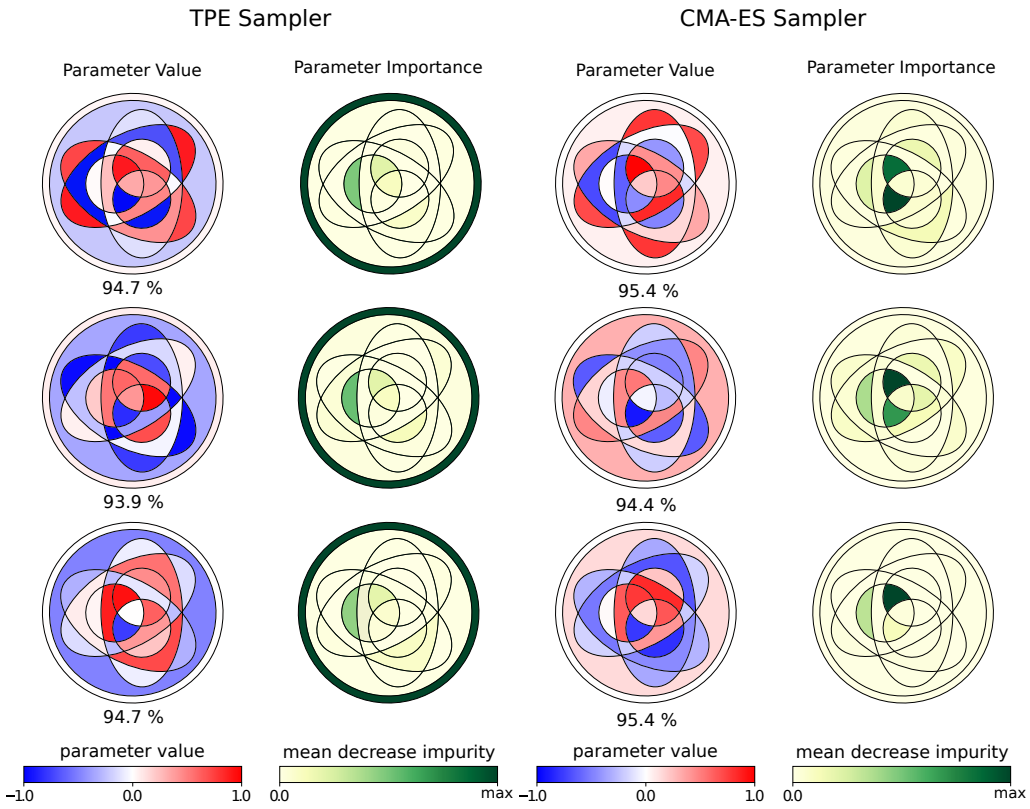

Figure 6: The three objective functions shown in the first column were obtained by utilizing the TPE sampler while the objective functions in the third column were obtained with the CMA-ES sampler. The performance illustrated below each of the objective functions corresponds to the maximum validation accuracy during optimization. The column next to the objective functions show the corresponding mean decrease impurity score of the corresponding goal parameter as a measure for the importance of the parameters. While the objective functions obtained with CMA-ES have a higher maximum validation accuracy than the goals obtained with the TPE, we observed that the first objective function function obtained with the TPE sampler (top-left) outperformed the others in terms of median validation accuracy for larger layer sizes, which is why we used this objective function for the main results.

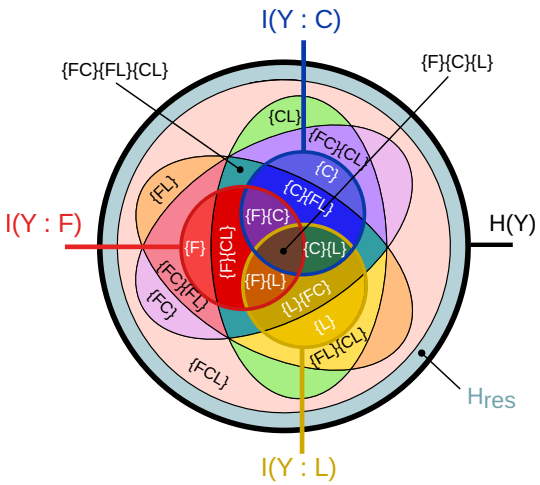

Figure 7: To improve the clarity and intuition for the atoms of the partial information decomposition with three source variables, this figure shows a larger version of Venn-diagram of the decomposition already shown in Fig. 1.**C**. The meaning of the different information atoms are also listed in Table 2 as a guide for the reader.

Table 2: A list of all atoms that are part of the Partial Information Decomposition with three source variables and their meaning.

Atom ($\Pi_{\text{Antichain}}$)	Meaning
$\Pi_{\{F\}\{C\}\{L\}}$	information that is redundant in all of the three sources
$\Pi_{\{F\}\{C\}}$	redundant information between feedforward and context input
$\Pi_{\{F\}\{L\}}$	redundant information between feedforward and lateral input
$\Pi_{\{C\}\{L\}}$	redundant information between context and lateral input
$\Pi_{\{F\}\{CL\}}$	information provided by both, the feedforward input and the synergy between the context and the lateral input
$\Pi_{\{C\}\{FL\}}$	information provided by both, the context input and the synergy between the feedforward and the lateral input
$\Pi_{\{L\}\{FC\}}$	information provided by both, the lateral input and the synergy between the feedforward and the context input
$\Pi_{\{F\}}$	unique information provided by the feedforward input
$\Pi_{\{C\}}$	unique information provided by the context input
$\Pi_{\{L\}}$	unique information provided by the lateral input
$\Pi_{\{FC\}\{FL\}\{CL\}}$	information redundantly provided by each of the pairwise synergies
$\Pi_{\{FC\}\{FL\}}$	information provided by both, the synergy between the feedforward and the context input and the synergy between the feedforward and the context input
$\Pi_{\{FC\}\{CL\}}$	information provided by both, the synergy between the feedforward and the context input and the synergy between the context and the lateral input
$\Pi_{\{FL\}\{CL\}}$	information provided by both, the synergy between the feedforward and the lateral input and the synergy between the context and the lateral input
$\Pi_{\{FC\}}$	synergy between the feedforward input and the context input
$\Pi_{\{FL\}}$	synergy between the feedforward input and the lateral input
$\Pi_{\{CL\}}$	synergy between the context input and the lateral input
$\Pi_{\{FCL\}}$	information encoded by all input sources synergistically

Table 3: The four most important non-zero goal parameters found through optimization on MNIST and ablation lead to interpretable requirements on each neuron's local information processing.

parameter	definition	value	interpretation of local information processing goal
$\gamma_{\{F\}\{C\}}$	redundancy between F and C	0.98	maximize information that is provided both by feedforward AND context but not by other neurons – and thus relevant for the task
$\gamma_{\{F\}\{L\}}$	redundancy between F and L	-0.99	minimize information that is redundant with other neurons, but NOT provided by the context – and thus not relevant for the task
$\gamma_{\{F\}\{C\}\{L\}}$	redundancy between F, C and L	0.33	moderately maximize information that is redundant with other neurons AND the context – thus relevant, but already encoded
$\gamma_{\{FC\}\{FL\}}$	redundancy between synergies requiring F	-0.97	minimize synergistic information that requires F and either C or L to be recovered

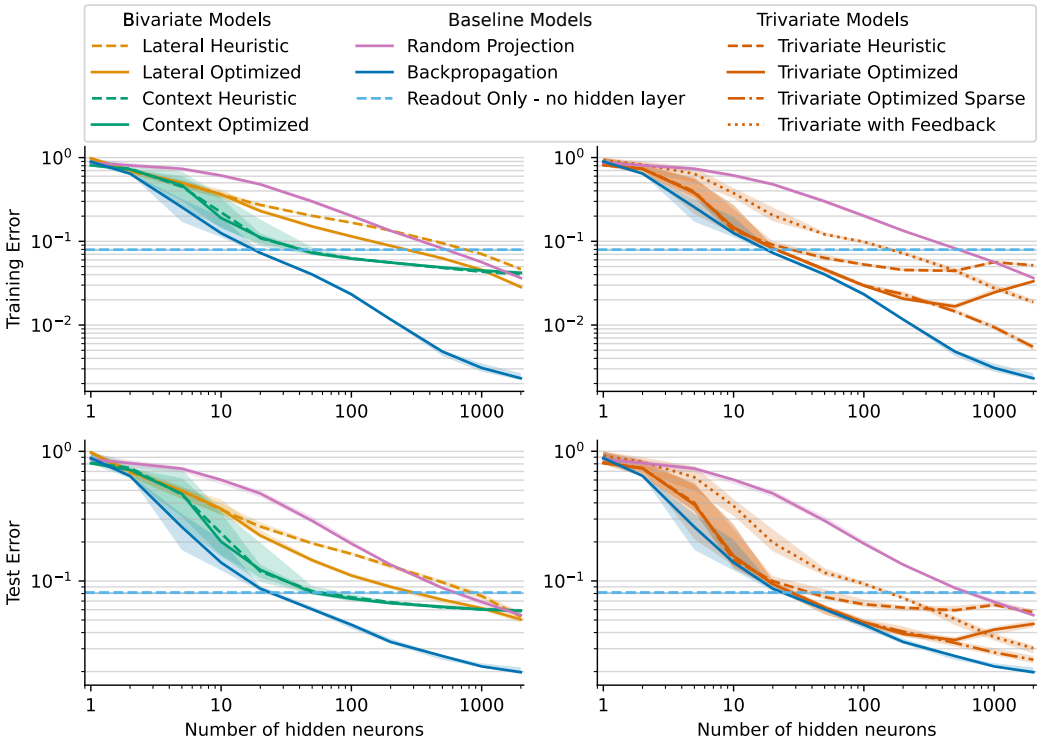

Figure 8: An overview over the training and test performances of the bivariate (left) and trivariate (right) models with different model structures and objective functions as well as the performances of some baseline models. In the bivariate models, only one of the non driving inputs (context or lateral) and the neurons optimized the heuristic objective functions illustrated in Fig. 1.**D**. The corresponding optimized objectives were optimized with the CMA-ES sampler similar to the trivariate objective function.

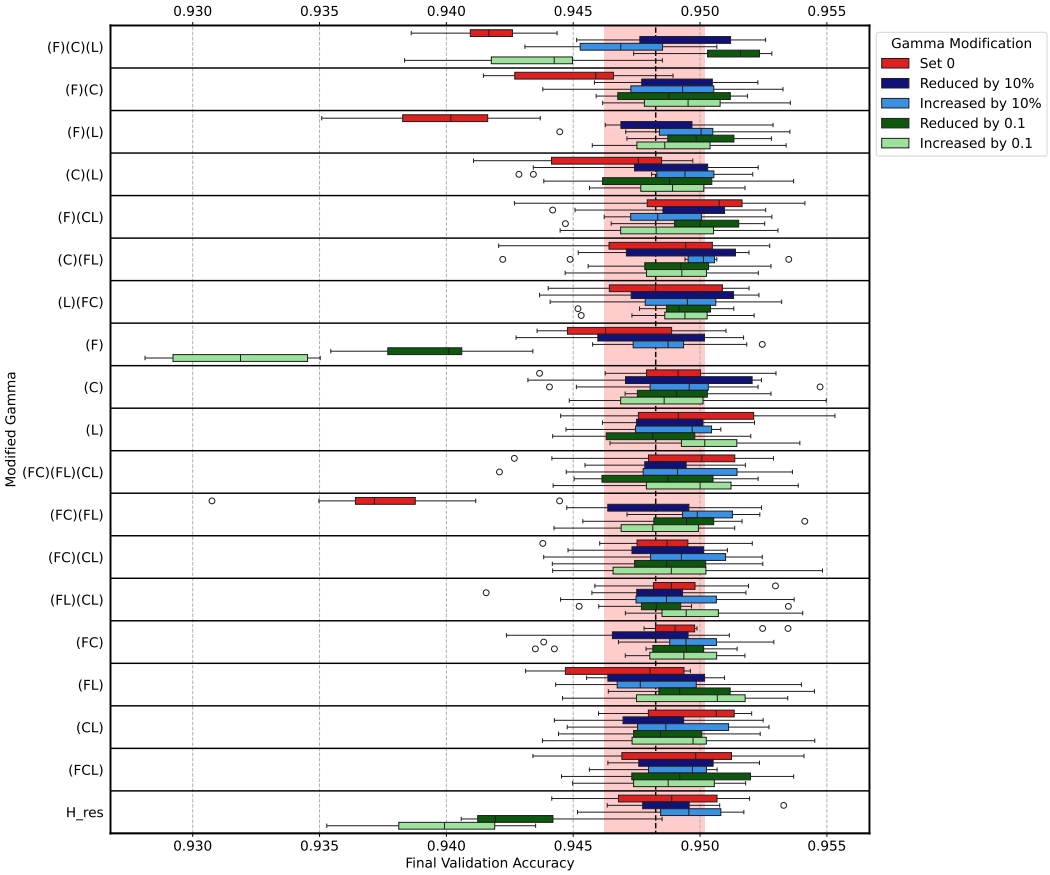

Figure 9: To estimate the importance of the different parameters of the trivariate objective function, we modified individual parameters of the trivariate objective function. The performance change induced by setting the parameters to 0 was already shown in Fig. 5.**B** and was used for the results shown in Fig. 5.**C**. Additionally, we modulated the parameters in an absolute and a relative manner to get an intuition about the importance of the fine tuning of the parameters.