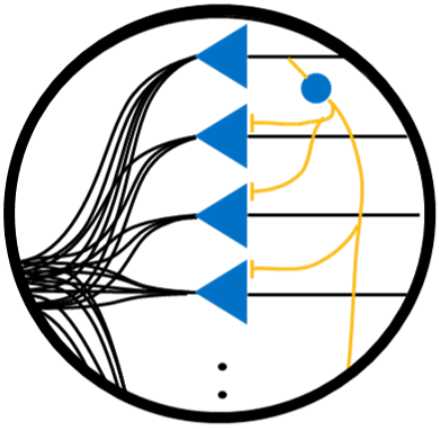
Intel Labs Announcement

Neuromorphic Computing Lab

Intel Neuromorphic Research Community

REQUEST FOR PROPOSALS (RFP)



Subject

Intel's Neuromorphic Computing Lab (NCL) seeks proposals for research projects that will help advance neuromorphic technology to real-world state-of-the-art applications. In this call, we are especially interested in foundational software contributions to the open source Lava framework as well as general algorithms and application demonstrations targeting Intel's latest Loihi 2 chip. We encourage all research findings to be published openly, with the goal of expanding the neuromorphic research community and accelerating the commercial adoption of neuromorphic technology.

Funding is available for eligible and suitably compelling proposals for up to three years in duration. We anticipate additional corporate and government sponsors (TBA) to consider funding projects relevant for their application domains based on their independent assessment of proposals submitted to this RFP.

Table of Contents

Key Dates

Upcoming INRC Talks and Information Sessions:

August 23, 2021 8:00-9:00am PDT. RFP Overview and Q&A.

August 30, 2021 8:00-8:30am PDT. RFP Q&A.

The August 23 session will be recorded and posted to the INRC website: http://neuromorphic.intel.com/.

Please email inrc_interest@intel.com for access to the INRC website and to receive an online meeting invitation if you are not already an engaged INRC member.

Proposal Submission Deadline (PIs):

For proposals requesting funding beginning EOY 2021: Oct 1, 2021

For all other proposals: We will consider access-only project proposals at any time.

The project proposal document template is available on the INRC website.

Overview

In 2018, Intel's Neuromorphic Computing Lab launched the **Intel Neuromorphic Research Community** (INRC). This collaborative research program is open to all academic, government, and industry research groups interested in exploring neuromorphic architectures for mainstream computing applications. In support of INRC projects, Intel makes our Loihi research chip and software available to members so the capabilities and advantages of neuromorphic approaches can be evaluated in a rigorous manner with real-world measurements and demonstrations.

In support of INRC research, Intel offers remote login access to Loihi systems and software development tools. Intel also loans physical hardware systems to teams that require physical access for their proposed research.

Intel hosts regular workshops and meetings open to INRC project participants to share results, discuss challenges, and provide hands-on training. A weekly online forum features presentations from Intel and INRC members sharing progress and new developments. Semi-annual week-long workshops bring the community together to meet, share progress, and discuss future directions. The INRC Winter 2021 virtual workshop attracted over 400 registrants.

The recent publication in Proceedings of the IEEE, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," [1] summarizes the findings of the first three years of research with Loihi. We expect new projects funded for ongoing research will incorporate these learnings and will focus on the most promising near-term directions for demonstrating commercial value.

To encourage growth of the community and convergence at the software level, Intel is launching the Lava software framework as an open source project on GitHub with permissive licensing (mostly BSD-3 and LGPL-2.1). Lava supports cross-platform execution, currently enabled on Loihi, Loihi-2, CPU, and GPU, and is available for porting to other platforms. Lava builds on a foundation of channel-based event-driven parallel processing with the goal of supporting a wide range of neuromorphic programming paradigms spanning deep learning to online learning to dynamics-based computing. Lava is modular, extensible, and incorporates most features from our previous NxSDK system. Projects supported by this RFP are expected to use, and preferably contribute to, the Lava framework.

Program Funding

To date, INRC projects have collectively received several million USD in funding from Intel's Corporate University Research Office (CUR) since 2018. Intel is now inviting proposals for 1- to 3-year projects that address one or more of the research vectors (RV) RV1 through RV5 outlined in this RFP. Based on available budget, Intel expects typical grants to support one student or postdoc per project. All proposals must justify the proposed budget in terms of the resources needed to carry out the proposed work.

Intel may share relevant submitted proposals with corporate and government members of the INRC interested in sponsoring research relevant to their application interests. To date three INRC projects have been funded by Accenture, and we anticipate broader support from corporate and government ecosystems in this round. Any group may opt out of this broader consideration by indicating so in their proposal.

Due to the limited grant funds available, we highly encourage researchers to leverage INRC support and membership to secure funding from other sources. Proposals outlining specific intentions to pursue such opportunities will be considered favorably. Feel free to submit requests for Intel letters of support at any time.

Background

Neuromorphic computing aims to apply insights from neuroscience to create a new class of computing technology that follows the form and function of biological neural networks. The goal is to discover a computer architecture that is inherently suited for the kinds of intelligent information processing that living brains effortlessly support.

Interest in neuromorphic computing has intensified in recent years due to several developments.

First, the success of artificial neural networks in the form of deep learning inspires confidence that biological insights can lead to great practical gains in computing and AI. While the breakthroughs coming from the deep learning approach are impressive and of tremendous practical value, deep learning models are facing limits in application scope because of their large data, power, and latency requirements.

Second, the golden era of process scaling that provided conventional architectures with steady and massive gains in computing power has passed. While process scaling continues to shrink transistor sizes, conventional CPU and GPU architectures struggle to use ever-increasing transistor counts to deliver commensurate gains in application performance and energy efficiency. This motivates new architectural approaches that can deliver greater application-level performance using smaller but slower circuits.

Finally, the pace of progress in neuroscience has accelerated dramatically in recent years, providing a wealth of new understanding and insights about the functioning of brains at the neuron level.

Neuromorphic computing represents a fundamental re-thinking of computer architecture at the transistor level. Compared to conventional architectures, it is massively parallel, with the fundamental unit of computation being a neuron with time-dependent dynamics, compared to processors executing sequential instruction streams in conventional architectures. The computation in the brain and in most neuromorphic algorithms is an *emergent phenomenon*, the result of collective interactions between simple neural units. In contrast, computation in conventional CPU, GPU, and matrix arithmetic

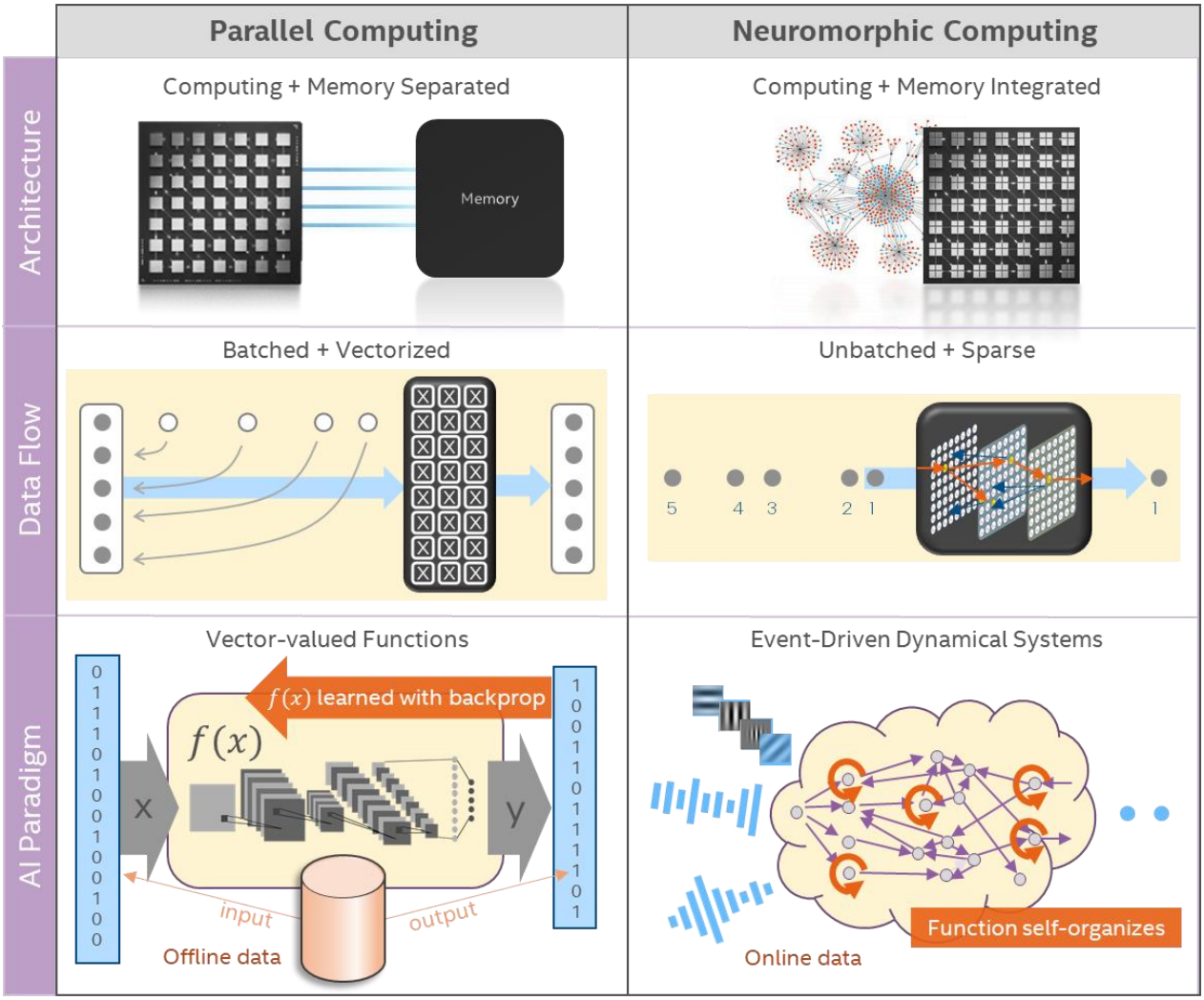


Figure 1. SIMPLIFIED VIEW CONTRASTING TODAY'S PARALLEL COMPUTING AND NEUROMORPHIC COMPUTING, WITH AI PARADIGMS CORRESPONDING TO EACH ONE.

processors is a precisely sequenced procedure accessing state from a shared address space. Communication in a neuromorphic architecture occurs in a peer-to-peer multicast fashion, with each neuron communicating scalar information to a diverse distribution of other neurons. Communication patterns are asynchronous, event-based, and extremely sparse in both time and space.

In a conventional processor, the top-level partitioning of the architecture between main memory and execution units leads to wide, vectorized datapaths that must stream data through the system at high bandwidth in order to achieve maximum efficiency. In a neuromorphic architecture, neural network weights and parameters are always stationary while only sparse data samples travel through the silicon. Data representations are low precision, often one bit, and all state changes, including weight changes, are the result of interactions between locally available quantities. Properties like noise, time-coding of information, and high-dimensional distributed data representations are used to achieve efficiency, robustness, and other surprising computational capabilities.

Figure 1 contrasts modern parallel computing and neuromorphic computing from an abstract architectural perspective. From this abstract view, ignoring for now any biological motivation, one can appreciate the bottom-up promise of the technology: low latency as a result of sparse, unbatched, and event-based data processing; resource-efficient processing of time-varying sensor input as a result of recurrent state updated locally per neuron, highly efficient online adaptation and learning as a result of fully localized state changes, and overall very low power as a result of its pervasive sparsity and activity-gating feedback paths. On the other hand, conventional parallel architectures supporting high precision matrix arithmetic are far better suited for offline training of differentiable and feed-forward models where sufficient pre-collected data is available.

As realized today by chips such as Intel's Loihi series [2], neuromorphic technology can provide value for applications characterized by specific properties, shown in Box 1. These loosely correspond to the same application needs that shaped brain evolution in nature. Rapid response to arriving information allows mobile organisms to evade threats, capture prey, while

Box 1. Application properties necessary for realizing gains on neuromorphic architectures compared to conventional computing architectures.

- **Streaming input data** with temporal information structure (e.g. audio, video, or any signals changing on microsecond-to-second time scales), especially when the data events of interest arrive infrequently and unpredictably.
- **A need for fast pattern matching, search, and optimization**. Neuromorphic architectures fundamentally implement a neural network computational model, which have a large body of literature supporting sub-symbolic processing and pattern matching in high-dimensional spaces, as well as optimizing network-defined energy functions through emergent neural dynamics.
- **A need for adaptation, fine-tuning, or associative learning** in response to arriving information.
- **A need for low latency responses**, e.g. as in closed-loop control applications. The time and resource cost of *batching* and *vectorization*, necessary for efficient use of conventional architectures, may be unacceptable.
- **Power constrained**. Often conventional architectures can achieve low latency at the expense of high power consumption. For suitable applications, neuromorphic architectures support both low latency and low power operation.
- **Relatively small problem scales** *or else* **cost insensitive**. Compared to conventional computing systems, neuromorphic systems contain a relatively small amount of aggregate memory, the result of its compute/memory-integrated architecture. While conventional architectures can scale to larger workloads by using large quantities of DRAM, neuromorphic architectures can only scale by instantiating more processing chips.

rapid online learning in response to minimal information allows organisms to respond to changing environmental conditions and outperform competition. Brain matter in nature is extremely expensive in both energy and material resources, just as we find in computing, so evolutionary pressures have led to designs that minimize resource consumption while maximizing behavioral objectives.

For these applications, Loihi has shown gains in latency and energy compared to conventional solutions into the orders of magnitude. These results are surveyed in [1] and summarized in Figure 2. Notable examples include constraint satisfaction, achieving up to 100,000x gains in energy-delay-product compared to conventional solutions.

Conversely, applications that do not exhibit the properties listed in Box 1 are unlikely to run better on neuromorphic architectures compared to conventional ones, at least not over the time frame of research projects funded by this RFP.

While a considerable body of results now exists pointing to the advantages of neuromorphic technology, the algorithmic methods and programming tools needed to realize this value for a wide range of real-world applications are still a severe limitation to progress. To enable commercially relevant applications and attract increased investment to the field, more attention needs to be directed to the most pressing of these near-term challenges. This is the objective of this RFP.

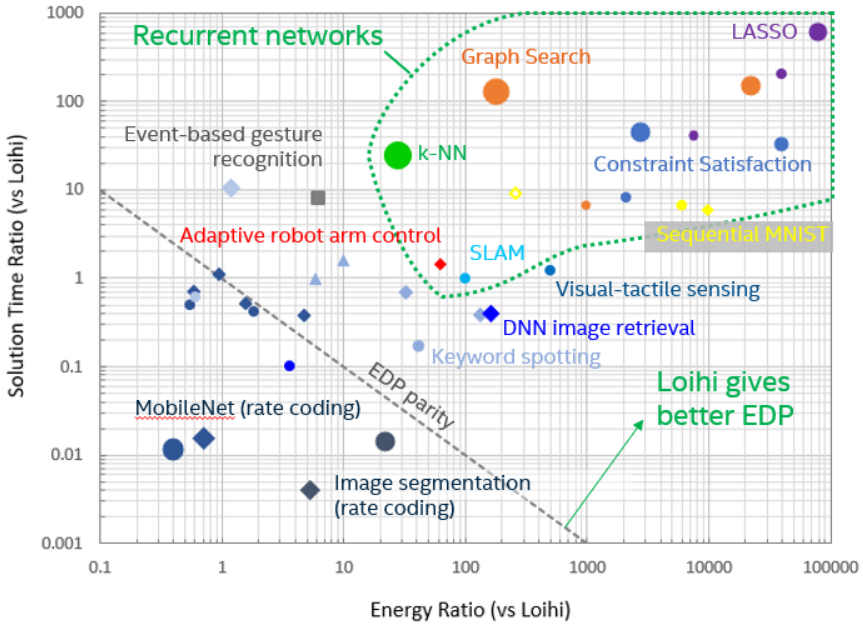


Figure 2. LOIHI RESULTS SHOWING RELATIVE GAINS IN SOLUTION ENERGY AND LATENCY VERSUS REFERENCE ARCHITECTURES. SEE (DAVIES ET AL 2021) FOR DETAILS.

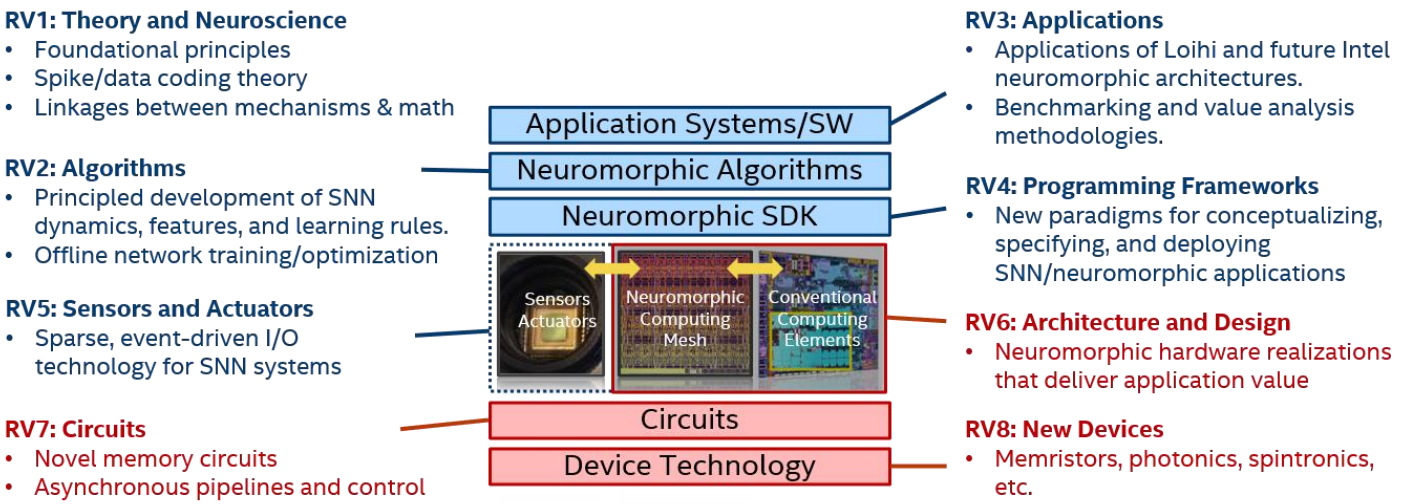


Figure 3. NEUROMORPHIC RESEARCH VECTORS. BLUE VECTORS FALL WITHIN THE DOMAIN OF INRC-FUNDED RESEARCH. RED ARE OUT OF SCOPE.

Technical Objectives of Research

Figure 3 defines the complete scope of neuromorphic computing research vectors. Although Intel pursues all of these vectors, we generally only consider funding external groups in vectors one through five, with an emphasis on vectors two through three.

Our interest at the present time is to focus funding in the most promising directions for near-term commercial value, detailed in the sections that follow. Intel funding will be directed to projects that align well with our prioritization or else present a strong case for near-term commercial relevance.

Proposals seeking funding should target Intel Loihi hardware platforms for final software, algorithm, and application demonstrations. This allows results to be rigorously quantified across all relevant metrics: correctness, precision/accuracy, speed of execution, power consumption, and resource utilization. Technical objectives should be defined in terms of those metrics such that results can be quantitatively assessed.

Groups seeking funding must articulate an informed, forward-looking orientation. Proposals should reflect a strong understanding of results obtained to date by the INRC or with other neuromorphic platforms, e.g. as documented in (Davies et al. 2021). Projects should not assume software programming or training capabilities that do not yet exist, or else they should present a credible plan for developing those tools in Lava and making them available to other INRC members.

Furthermore, projects should anticipate the future direction of neuromorphic hardware architecture evolution, e.g. as will be imminently available in Loihi 2. A few of these new directions are provided in Box 2.

Our broader objective is to accelerate progress in a collaborative fashion, so we wish to see functional code contributed to the Lava GitHub with permissive licensing terms. This allows others to replicate and advance on progress.

RV1: Theory and Neuroscience

The long-term adoption of neuromorphic technology depends on establishing sound theoretical foundations that support robust algorithms and applications. As a new computational paradigm differing from conventional computer architectures in fundamental ways, neuromorphic computing currently lacks unifying theoretical frameworks such as the Turing Model, which leads to a highly fragmented exploration space.

Past INRC RFPs have highlighted the need for theoretical work in the areas of computational complexity frameworks, characterizations of neural dynamics, and unified engineering figures of merit. While these areas remain important for progress in the field, they are not areas we expect to support with Intel funding at this time.

Instead, we seek to fund areas of theoretical research focused on removing practical roadblocks to neuromorphic technology deployment. These include:

- Stability guarantees for (spiking) neural network adaptive control algorithms.
- Frameworks that provide principled motivation for different information coding strategies based on application objectives. For example, when to use temporal coding, phase coding, rate coding, population coding, plus generalizations to other coding strategies such as sigma-delta with graded spikes.
- Neuroscience basis for stochastic spiking neural networks (SNNs). Adding noise to SNNs has provided some of the most compelling computational results on Loihi to date. Establishing a linkage between the mathematical formulation of these models [3] and more bio-plausible neural circuits could provide insights for how to enhance these networks (e.g. with plasticity) and how to integrate them with more conventional networks.
- Neuroscience foundation of vector symbolic architectures [4] and linkage to dynamic neural fields [5].

Box 2. Future directions of Intel's neuromorphic hardware architecture.

- **Graded spikes**. Many traditional SNN chips, including Loihi, only support binary-valued spike messages. While binary spikes can perform a remarkable amount of computation (as best demonstrated by the brain), in digital hardware spikes can be easily generalized to carry integer-valued payloads with little extra cost in either performance or energy. Several recent neuromorphic chips support such graded spikes, and future Intel chips will too. These generalized spike messages support event-based messaging, preserving the desirable sparse and time-coded communication properties of SNNs, while also providing greater numerical precision.
- **More general neuron models**. Loihi supports a generalized leaky-integrate-and-fire (LIF) spiking neuron model with the ability to aggregate neural units into dendritic trees communicating graded dynamic state variables towards the soma (root) compartment. Intel's future neuromorphic chips will generalize this further, with fully programmable neuron models that allow each compartment to state variable changes as nearly arbitrary difference equations with configurable spike conditions and state machines. A greatly expanded set of neuron models will be supported, including adaptive threshold LIF, Resonate-and-Fire, Hopf resonators, sigma-delta coding, and many others.
- **Three-factor learning rules**. Loihi primarily supports two-factor learning rules (involving pre- and post-synaptic traces), with a third modulatory term set in a diffuse manner from graded "reward" broadcasts. Future chips will support more targeted and localized third factors in learning rules, for example those mapped to specific postsynaptic neurons as projected error signals. Neuron thresholds and other parameters will also support more flexible programmed plasticity.

RV2: Algorithms

Central to the advancement of neuromorphic computing is the development of algorithms that leverage the novel features of neuromorphic architectures, while satisfying their bio-inspired hardware constraints. These are algorithms that utilize *sparsity* of connectivity, communication, and activity. They should include *dynamically evolving state* within each neuron that is excited by inputs and inhibited through feedback loops. Learning algorithms can only use state variables that are *locally available* at each synapse and neuron. Novel neuro-inspired features such as stochasticity, structural plasticity, event-driven computation, and temporal information coding can also provide unique gains on neuromorphic architectures.

Algorithms proposals should fully consider all recent learnings [1] and should include a plan for rigorous benchmarking to current state-of-the-art conventional solutions. The value of the proposed algorithms should be motivated in the context of a specific application and associated real world constraints, informed by the challenges and opportunities facing neuromorphic technology deployment.

RV2 projects should advance beyond theory and modeled examples to provide generally usable software modules in Lava targeting Loihi and other future neuromorphic platforms. Others should be able to easily apply results to their own problems, preferably over a range of different application domains.

The following sub-vectors are of particular interest.

RV2.1 Novel neuromorphic neuron models

Compared to the simple stateless neuron models of deep learning (e.g., ReLU), biologically inspired neuron models include *time-varying state variables*. In neuromorphic hardware, these offer several computational advantages:

- They introduce time-varying behavior into a network that can otherwise only be achieved through iterative evaluation of the entire network.
- By combining a diversity of such neurons in a network, each evolving its state variables independently of the others without the need for network-wide communication, the network can transform inputs to a sparser form with a richer set of time-varying basis functions. (For example, an array of resonate-and-fire neurons with different intrinsic frequencies can approximate the Short-Time Fourier Transform.) This can reduce network size and parameter counts compared to homogenous networks.
- Stateful neurons can predict future inputs by using their stored state, allowing them to communicate only unanticipated changes. This can lead to sparser activity (e.g. CUBA LIF and Sigma-Delta neurons).

Loihi implements a single generalized LIF model that can be parameterized and aggregated into dendritic tree-like structures as a model of complex neurons. We are now interested in supporting research into a broader class of neuron models that further generalize Loihi's (See Box 2.) These models can have nearly arbitrary internal dynamics and support both spike-based communication and continuous transmission of graded information. The former sparsifies long-range communication with event-based messages triggered by some spike condition; the latter provides high-precision computation within a local cluster of neurons, where the cost of communication and fanout are low.

Potential neuron models of interest include oscillatory models (resonate-and-fire, Hopf oscillators), sigma-delta coding models, stochastic models, and hybrid combinations of stateless nonlinear dendritic structures with stateful soma units.

Generally, we see near-term promise of this algorithmic approach for demonstrating more compact, intelligent, and

efficient nonlinear signal processing solutions, e.g., for audio processing.

RV2.2 Optimization with neurodynamics

The best quantified Loihi results come from networks that optimize well-defined objectives using attractor and other network-level dynamics. Examples include Lasso regression, constraint satisfaction, and similarity search, with other promising generalizations on the horizon such as Subspace Locally Competitive Algorithm [6], Minimax optimization [7], genetic algorithms [8] and probabilistic inference [9] [10] [11].

One pressing challenge in this domain is to bridge between continuous, time-varying input signals and the episodic nature of current neuromorphic solvers. For example, while Loihi can solve Lasso problems with incredible speed and efficiency using SNN dynamics, solutions can be brittle, varying greatly from one input sample to the next (perhaps only due to noise), which limits its value in a real-time setting. However, as standalone offline solver, its runtime is dominated by I/O.

Another important challenge involves hierarchically composing such optimizing networks to solve larger problems or to solve the same problems with higher precision. Multiscale modeling techniques from conventional applied math may be of value here, in addition to novel insights from neuroscience.

RV2.3 Online learning

In the neuromorphic research field at large, much attention and funding has been directed to learning algorithms that approximate backpropagation (or gradient descent in parameter space) with online learning rules that respect locality and other neuromorphic architectural constraints [12] [13] [14] [15]. Despite encouraging progress, major hurdles remain to be resolved before these approaches can yield practical value. While these algorithms can, in their execution, operate online, they rely on unrealistic assumptions about the statistics of real-world data (namely independent identically distributed (iid) samples). Furthermore, merely operating online doesn't improve the data efficiency of backprop, which is a fundamental challenge with this general category of learning for online/edge applications.

We are interested in supporting novel learning algorithms that comprehensively address the requirements for online deployment. (1) continuous/lifelong learning approaches that naturally avoid or manage catastrophic forgetting with extra readily integrated ingredients; (2) more data efficient learning algorithms that achieve excellent data efficiency by leveraging pre-training, hierarchy, causality, or priors in a principled manner; and (3) algorithms that utilize recurrent network structures and dynamics in order to deliver maximum execution efficiency on neuromorphic hardware.

Several example approaches of interest are described below.

On-chip few-shot transfer learning. Few-shot on-chip learning (last or last few layers) on top of pre-trained feature extraction layers needs more attention. E. Neftci's Surrogate-gradient Online Error-triggered Learning (SEOL) work is a good starting point [16]. There is scope for work on improving the on-chip learning behavior. Offline training with Model Agnostic Meta Learning (MAML) for better feature extraction layers which are better suited for few-shot adaptation for the task at hand needs investigation for sparse-event based computation. MAML is an established idea for standard ANNs, but adaptation to spike/event-based computation has not yet been pursued.

Contrastive or surprise-driven learning. In real world scenario, it is difficult to find good labelled data available. A form of continual unsupervised/semi-supervised learning learns with pseudo-labels continuously and typically results in general features agnostic to the final task, therefore, the resulting features are more generalizable. Tasks may be formulated to contrast between similar and dissimilar instances leading to good latent representations in a semi-supervised manner. Robust continuous learning may be achieved by contrasting the network's internal prediction of activity and actual activity (a form of surprise). Examples include CLAPP [17], Contrastive Predictive Coding [18], and MoCo-v2 [19].

Stochastic learning with bounded resources to avoid catastrophic forgetting [20] [21]. These approaches may use multi-state synapses and their internal dynamics [22] [23] [24] [25], complex synapses to extend memory lifetime with optimal synaptic model [26], or *replay* for pinning significant memories [27] [28] and use them for key retrieval of in between memories [29]. The concepts of bounded resources and multi state synapses align well with the features and constraints of neuromorphic hardware.

Hierarchical Reinforcement Learning (HRL) for reducing state space dimensionality and learning complexity [30] [31] [32] [33] learning the hierarchy structure with solving the HRL problem [34] and introducing transfer learning to speed learning of novel/similar complex structured memories.

Shallow associative learning utilizing high dimensional attractor dynamics in conjunction with neuro- or synapto-genesis, e.g. olfaction-inspired approaches [35].

Causal representation learning has emerged at the intersection of DL and probabilistic graphical models to learn causal representations from few samples that are robust to distribution changes and thus generalize well [36]. This is achieved by going beyond learning mere correlations from iid samples towards incorporating counterfactual queries or active interventions while sampling to learn factorized models. Through their modularity, such models promise local and efficient updates of sparse causal relationships as opposed to distributed updates of acausal models and thus offer a path to robust continual online learning. While the mapping of such graphical models and associated learning rules to neuromorphic HW is not completely clear yet, these models capture some of the fundamental characteristics of spiking neural networks like causal event-based operation, sparsity, active sensing, and embodying probability distributions with stochastic spiking dynamics [9].

RV2.4 Stochastic spiking neural networks

Stochastic spiking neural networks have been used to represent Bayesian or more general graphical models [9] and are known to solve a wide range of hard problems such as the computation of

marginal probabilities, or maximum likelihood [10]. These models have for instance been applied to solving constraint satisfaction problems with SNNs [37] and could offer one possible realization of causal graphical models for efficient continual learning. Benchmarked recently on Loihi hardware, such stochastic spiking networks are now demonstrating significant outperformance compared to classical approaches on conventional hardware architectures [1]. We wish to encourage greater focus on this class of networks. We see an opportunity for rapid progress on several fronts: (1) maturing previously proposed theory and fully demonstrating these frameworks on neuromorphic hardware, (2) combining such stochastic networks with more standard components (e.g. offline-trained DNNs and associative memories), and (3) algorithmically extending these frameworks to related areas of high value such as probabilistic inference and model predictive control.

RV2.5 Vector Symbolic Architectures

Vector Symbolic Architectures, also known as Hyperdimensional Computing algorithms, are attracting increasing interest [4] and advancing with compelling new algorithms such as the resonator network [38]. While promising, VSA application demonstrations to date remain small scale and future complex scaled up VSA algorithms utilizing resonator-like elements face significant compute challenges due to their heavy use of high-dimensional associative memories and attractor dynamics.

Research into VSA frameworks that bring sparsity to the connectivity and activity of VSAs may enable this class of algorithms to be efficiently mapped to neuromorphic hardware and far outperform conventional dense-vector approaches. Insofar as the brain implements such hyperdimensional vector symbolic computations, we can be confident that a highly efficient, neuro-inspired sparse implementation exists. Promising progress on sparse binding has already been demonstrated [21] and we encourage greater attention to this promising new direction for neuromorphic computing.

RV2.6 Offline training and network optimization

While a wealth of backprop-style offline training tools for spiking neural networks have been demonstrated over the past few years, severe computational constraints limit them all to small-scale applications. Scaling up computational resources quickly faces diminishing returns. Therefore innovations in offline optimization methods are needed to successfully train larger and more complex neuromorphic networks. Innovations might include hybrid training approaches (e.g. ANN conversion-based pre-training followed by direct fine-tuning), Hessian-based methods, novel neuron models and features to ease training, among others.

Meanwhile, we see a clear fundamental limit to the value of backprop-based offline optimization for neuromorphic networks, which are inherently non-differentiable. Following nature's lead, we see a promising future for evolutionary methods for long-term neuromorphic algorithm discovery and optimization. These approaches are currently at a very early stage, showing even less success so far in scaling compared to backpropagation approaches. We are interested in supporting new ideas and approaches that can credibly promise significant gains.

RV3: System Applications

We seek commercially relevant application demonstrations at the intersection of research and today's best engineering solutions. Compared to past application demonstrations with Loihi, we now intend to raise the bar of anticipated impact. This narrows the scope of candidate opportunities. Over the long term we see a vast domain of applications for neuromorphic devices, but in the near term, we see a more limited list of candidates that are commercially relevant yet viable using today's neuromorphic hardware and known algorithmic methods.

To substantiate commercial relevance, we prefer application projects that include participation, support, or co-investment from industry or government organizations. For example, this could be a corporate advisor, use of data from an end customer, or integration into a commercial system platform. For suitably compelling proposals, Intel is open to partnering and contributing engineering resources if doing so will support a successful outcome.

The technical suitability of application proposals will be evaluated according to the criteria included in the INRC project proposal template, which is included as Appendix 2 here. A comprehensive, convincing response to these questions is the single most important factor in our evaluation of application proposals. The algorithmic risk in the proposed solution should be relatively low; solutions should use algorithmic methods that have been previously published and preferably already validated on neuromorphic hardware.

All application proposals should clearly describe the solution's value in relation to limitations of current solutions, and how specifically the neuromorphic approach is uniquely capable of outperforming. For projects in this RV, domain expertise is as important if not more important than a background in neuromorphic research. Proposals should articulate a strong case for the commercial viability of a successful outcome, based on measurable and significant advances the project will demonstrate over the current state-of-the-art.

Some promising categories of application demonstrations are listed below.

Audio processing, especially applications that need to operate continuously in an always-on fashion at low power levels and where a fast response and online adaptation is needed, e.g. wake-on-voice, dynamic noise suppression, automatic speech recognition, sound detection and localization, speaker identification, and blind source separation.

Signal processing for security, failure detection, and sensor networks. Examples range from radar, sonar, biometric, and turbine monitoring to cybersecurity intrusion detection to sensor network processing for earthquake prediction and oil field analysis.

Human-machine interfacing: gesture recognition for cursor control or sign language interpretation, gaze tracking, speech

Box 3. What is Lava?

- **Magma**. At its foundation, replacing the previous "NxSDK", is Intel's next generation SDK that provides a low-level interface for mapping and executing neural network models and sequential processes (snips) to neuromorphic hardware. This layer now includes cross-platform execution support so applications can be developed in simulation on CPU/GPUs before being deployed to Loihi (or other) neuromorphic platforms. This layer also includes a profiler that can measure or estimate performance and energy consumption across the targeted back-end platforms.
- **Channel-based asynchronous message passing**. Lava specifies, compiles, and executes a collection of processes mapped to a heterogenous execution platform including both conventional and neuromorphic components. Unifying all inter-process communication is an event-based message passing framework sometimes referred to as Communicating Sequential Processes or the Actor model. Messages in Lava vary in granularity from single-bit spikes to buffered packets with arbitrary payloads.
- **Offline training.** Lava supports tools such as SLAYER [44] so a range of different event-driven neural networks (LIF SNNs, RF SNNs, SDNNs, and others) can be trained offline with backpropagation and integrated with other modules specified in Lava.
- **Integration with third party frameworks**. Lava is fully extensible, supporting interfaces to third party frameworks such as ROS, YARP, TensorFlow, and hopefully many more in the future, providing a truly heterogeneous execution environment.
- **Python interfaces**. For ease of adoption, all libraries and features in Lava are exposed through Python, with optimized libraries and C/C++/CUDA/OpenCL code under the hood where necessary to provide excellent performance.
- **An open-source framework with permissive licensing**. Lava is hosted publicly on GitHub and runs on CPU/GPU platforms without requiring any legal agreement with Intel. The software is available for free use under BSD-3 and LGPL-2.1 licensing.

processing, tactile/haptic sensor processing. Brain-computer interfaces (EEG, EMG, direct nerve/neural probes) that demonstrate real-world advantages for gaming and people with disabilities. Additional value may come from applying neuromorphic compute to wireless interfaces in this domain.

Visual-inertial odometry and SLAM, in particular projects that build on past SLAM demonstrations with Loihi [39] [40] to credibly advance state-of-the-art odometry, localization, and mapping capabilities in edge devices such as Intel's RealSense cameras.

SWaP-constrained visual inference, learning, and control, e.g. for satellites, drones/UAVs, and robots, calling for fast, online adaptation to unpredictable environments and new object classes. Note that in the near term, the complexity of these visual inference problems is limited by neuromorphic system capacity. Therefore, straightforward DNN translation methods are inadequate, and we will be looking for projects that apply the latest algorithmic insights for maximizing resource efficiency (e.g. novel neuron models and sparse networks) while minimizing energy/latency (e.g. by leveraging attractor dynamics).

Optimization for real-time decision making and control applications. Lava libraries will be available soon enabling Loihi platforms to solve a range of constrained optimization problems 10-100x faster and >1000x more efficiently than conventional methods. These include constraint satisfaction, mixed-integer linear programming, quadratic programming, and QUBO. We are interested in identifying applications where these gains provide the greatest commercial value while satisfying the limitations of neuromorphic solutions (namely low precision variables, quasi-static problem structure, and limited I/O bandwidth).

RV4: Programming Models and Frameworks

Despite many recent demonstrations of compelling neuromorphic capabilities, the level of overall software maturity in the field remains low. Code sharing between groups is almost nonexistent, and published examples generally are difficult if not impossible to replicate by others. There are very few examples of composability, abstraction, and modularity in the algorithms studied and published. While some promising frameworks have open-source code, prohibitive licensing terms limit widespread adoption and community-wide contribution.

Intel is attempting to improve this area in a variety of ways, such as by providing the Nx Software Development Kit free for INRC use, hosting a GitHub space to share INRC project code, funding promising software toolchains such as Nengo, and now launching the **Lava framework** that we hope will encourage convergence on a single cross-platform, extensible, and open neuromorphic software framework. (See Box 3 "What is Lava?")

With this RFP, we hope to focus attention and funding across all levels of a single unified software framework to help resolve these challenges. The foundational SDK ingredients that Intel has already released for open-source development support Intel's Loihi and future neuromorphic chips, as well as execution and offline training on CPU/GPU, and may be ported to other platforms. Building on this foundation, we hope to encourage developers with diverse backgrounds and interests to help improve and extend the Lava framework. By bringing new ideas

and perspectives to the software challenges, we see opportunities for great gains in many areas that are bottlenecks to progress today: developer productivity, offline training speeds, module composability, hardware mapping speed and optimality, and libraries for powerful features like structural plasticity and evolutionary optimization.

Beyond the immediate priority of building out and optimizing the central capabilities of the Lava framework, we see several long-term compelling directions for Lava development, listed below. We would welcome the academic community to take the lead in these areas, with the assistance of Intel funding and partnership.

Development of a Domain-Specific Language (DSL) spanning all levels of the underlying heterogeneous compute platform and abstractions. The goal would be to unify these elements with a common asynchronous message-passing language following the model of Communicating Sequential Processes [41] and Actor-based languages (e.g. Go, Akka, Lingua Franca [42]) yet incorporating the unique properties of neuromorphic computing that don't conform to a von Neumann programming model such as collective dynamics, differentiable programming, stochasticity, and plasticity.

Event-driven Actor Virtual Machine (AVM) for CPU-based execution. The current Lava backend simulator for CPU/GPU leverages TensorFlow, which is not optimal for executing the sparse and event-driven applications for which Lava is intended. It also fails to unify the execution of both neural processes and conventional sequential ones in a single runtime system. See FNS [43] for a recent SNN-specific simulator example.

Formal specification and verification of Lava processes spanning structural, functional, and behavioral levels of abstraction. A promising approach for improving the explainability and composability of certain classes of neuromorphic networks is to rigorously derive emergent properties of their dynamics, e.g. attractor states and stability conditions, given a specification of their structure and parameters. Such capabilities would abstract away low-level details of the networks' behavior, such as precise trajectories in phase space, while providing practical insights to the application developer. Such neuromorphic network verification capabilities would require new mathematical analysis tools going beyond the discrete math and logical theorem provers of conventional formal software verification.

RV5: Event-Based Interfaces

Over the past several years, event-based vision sensing technology has seemingly matured with the advent of commercially available sensors and large investments from numerous industry and government organizations. Neuromorphic processing of event-based sensor output promises many advantages over conventional architectures, yet algorithmic and hardware scaling challenges limit the near-term commercial viability of this combination of technologies. We view some of these challenges as fundamental, exacerbated by the pixel-level granularity of features produced by today's event-based sensors.

We are interested in supporting fundamental research that addresses these pain points: (1) application-driven modeling of future sensor architectures that tightly integrate novel photodiode sampling circuits with neuromorphic processing, both near and far; (2) novel spatiotemporal filtering techniques prototyped on Loihi and/or FPGAs that extract meaningful features with a minimum of parameters and compute cost; (3) feedback-driven attention and active sensing mechanisms that improve the speed, efficiency, and resource needs of visual inference and learning.

Beyond vision sensors, Intel may consider funding and offering in-kind support for interface and hardware engineering projects demonstrating the value of novel event-based sensor and actuator technologies, such as electronic skins, cochlea-inspired audio processing, muscle-like actuators, and wireless interfaces.

Overall Research Goals

Intel's goals for the INRC relate to accelerating neuromorphic research progress and enabling the commercial adoption of future neuromorphic technology:

- Identify, develop, and characterize algorithms that exploit the novel properties of event-based/spiking neuromorphic hardware architectures to deliver orders of magnitude gains in latency, energy efficiency, and data efficiency compared to conventional solutions.
- Guide the iterative development of neuromorphic, non-Von Neumann architectures with the algorithmic and architectural insights from the above.
- Develop an open, cross-platform software framework that aligns and grows the neuromorphic research ecosystem, laying the groundwork for a future commercial developer ecosystem.
- Prototype real-world applications with Lava and Intel's neuromorphic research silicon to identify the areas where neuromorphic technology might deliver the greatest commercial value.

In its funding decisions, Intel will prefer project proposals that align well with these goals.

Engagement Process

The process for formally joining the Intel Neuromorphic Research Community is outlined below.

1) Complete and submit one or more INRC project proposals
2) An *INRC participation agreement* needs to be executed between Intel and the institutions affiliated with all team members, once the proposal is accepted. An *authorized legal representative* from each institution needs to sign this agreement. This is typically not the PI. Specific terms of the agreement may be negotiated in some cases, when necessary. PIs are responsible for ensuring all project team members are aware of applicable terms of the INRC participation agreement.

3) Once a project has the green light to proceed, members of the project formally become INRC participants. This authorizes access to the following services:
    a. Engaged member sections of the INRC website, including detailed Loihi documentation and content contributed by other INRC members.
    b. Remote access to Loihi via our Neuromorphic Research Cloud (NRC) system.
    c. Access to the Loihi-proprietary portions of the Lava SDK ("*Magma*"), necessary for running Lava applications on Loihi hardware platforms.
    d. Physical access to loaned Loihi hardware systems (as needed/approved)
4) PIs or their delegate should create a project page in the "Projects and Results" space of the INRC website, which provides a single location for maintaining the team's up-to-date member list, results, and other resources of general INRC interest.
5) Conduct your research project.
6) Share progress, results, and demonstrations in periodic INRC online forums and face-to-face workshops open to all INRC members.
7) At the conclusion of the project, present a final report/demo, publish results, and contribute as much software as possible to the Lava GitHub site.

Having successfully completed at least one INRC project, you remain an INRC member but, due to capacity constraints, you may lose access to Loihi/NRC resources unless approved for another project.

Additional information describing the engagement process are available on the INRC website. Further guidance will be provided upon receipt of a project proposal. Feel free to send questions at any time to inrc_interest@intel.com.

Eligibility

We welcome groups of all types and locations to submit research proposals and engage in the community, subject only to U.S. export control laws.

Intel research grants are only offered to academic research groups.

Project Proposal Submission

Along with this RFP, we are providing an *INRC Project Proposal Template* document that lists all information and documentation required of each respondent. Please refer to that document for detailed guidance on what information to include in your INRC project proposal, while preserving the template structure. A proposal may be rejected if it does not include the required information and documents.

Some recommendations:

- Delete all commentary and guidance text from the template document.
- Strive for brevity.
- Feel free to submit more than one proposal.
- Keep the scope of each project narrowly defined and limited to a single research vector.

Please note that we are unable to receive proposals that are provided under an obligation of confidentiality. Proposals should therefore include only public information. If you represent a corporate entity with proprietary IP considerations, please contact us prior to submitting a proposal.

Progress Updates and Results

Once your INRC research project is approved and underway, we would like to receive quarterly updates on your progress and also notification of any work published as a result of this research. If you receive Intel funding, we will expect progress updates at least once per quarter, with at least one per year at an INRC face-to-face workshop.

Progress updates will generally be in the form of short ~30 minute presentations, open to all other INRC participants. Live demonstrations of results using Loihi, emulation, or simulation when possible will be highly appreciated.

Intel supports and encourages publishing results in public, peer-reviewed forums. We will do our best to support live demonstrations and independently hosted hands-on workshops using Loihi hardware systems, subject to U.S. export control, security, availability, and other constraints.

Loihi Hardware and SDK

Intel offers a wide variety of Loihi-based neuromorphic systems and software for you to utilize in your research.

Neuromorphic Research Cloud (NRC)

Our primary means of providing you access to Loihi is through our Neuromorphic Research Cloud. Approved projects will be given their own virtual machine on the cloud with SLURM access to either our standalone 32-chip Nahuku system or partitions of multiple Nahuku-32 boards within our large-scale Pohoiki Springs system. We will be adding single-chip Loihi 2 systems to the NRC soon, with multi-chip systems following in the coming months.

Nahuku-32

For some projects, the NRC may not work for you. Instead, you may need local access to our Nahuku-32 platform. While we have a limited supply, if you provide clear reasoning within your proposal for one, we can loan these systems for up to 3-6 months at a time. We have a smaller variant as well (Nahuku 08) which can be loaned for longer periods.

Similar, but more compact, Loihi 2 systems will be available for local lab use in 2022.

Kapoho Bay

Another system available for loan is our Kapoho Bay USB form factor. It supports up to 2 Loihi Chips and is compatible with Ubuntu 16.04 and 18.04. Great for portable projects and equipped with GPIO pins for DVS camera input, this device can be loaned to project teams for up to 1 year.

Lava software framework

Lava is freely available on GitHub for all to encourage community growth and convergence. See Box 3 for its major defining ingredients. Lava replaces and expands on our previous NxSDK software that was only available to engaged INRC members. The lowest level components necessary for deploying applications to Loihi hardware systems remain accessible only to engaged INRC members, at no cost.

Evaluation Criteria for Proposals Seeking Funding

In order of importance, the evaluation criteria for this solicitation are as follows:

1. **Potential contribution and relevance to Intel and the broader industry**: The proposed research should directly support a technology solution that addresses the RVs outlined above, leading to technological advances with the potential for ongoing technology transfer in collaboration with Intel and the broader industry.

2. **Technical innovation**: Proposed solutions of interest should clearly push the boundaries of technical innovation and advancement. Research that is not of interest in this program include incremental advancements to state-of-the-art and current design practices. Feasibility of new algorithms/techniques should be demonstrated through SW/HW implementations. Projects seeking funding should target Loihi hardware platforms and the Lava software framework to enable algorithmic capabilities and application proof of concept demonstrators that others can build on. Technical objectives should be defined in terms of quantitative target metrics (precision/accuracy, speed of execution, power consumption, and resource utilization) as detailed in "Technical Objectives of Research." Funded projects will be enabled with remote access to Loihi and future neuromorphic hardware platforms via our Neuromorphic Research Cloud (NRC) system and limited access to physical loaned systems as needed/approved. See sections on Engagement Process and Loihi Hardware and SDK for more information.

3. **Clarity of overall objectives, intermediate milestones and success criteria**: The proposed Research Plan should clearly convey that the PIs have the knowledge and capability to achieve the stated research goals. It is understood that any research program will have uncertainties and unanswered questions at the proposal stage, but a clear path forward in key challenge areas must be identified and justified. Teams are expected to demonstrate progress toward project goals at quarterly milestones and monthly project status updates. The proposal should explicitly point out which RV is being addressed, the synergy among them if more than one RV, the plan and milestones towards building research prototypes, plan for ongoing technology transfers, and the anticipated proof of concept outcome. The technical suitability of proposals to RV2:

Algorithms and RV3: Systems Applications will be evaluated according to the criteria included in the INRC project proposal template, as included for reference in Appendix 1 and Appendix 2, respectively. Strength of project management will also be considered.

4. **Qualification of participating researchers**: The extent to which expertise and prior experience bear on the problem at hand. Please elaborate on track records of building research prototypes (e.g., open-source research code/collaterals on GitHub) and resulting publications from past relevant projects.

5. **Cost effectiveness and cost realism**: The extent to which the proposed work is both feasible and impactful within the proposed resource levels will be examined.

6. **Potential for co-funding**: Opportunity for closely synergistic matching grants and co-funding with other funding entities, such as SRC, NSF, DARPA, NSERC, etc. will be given significant consideration.

7. **Potential for broader impact**: Intel supports the advancement of computing education and diverse participation in STEM. Significant consideration will be given to proposals in which the outcome of the research can influence the development of new curriculum initiatives impacting undergraduate or graduate education at the respective universities (e.g., exposure to latest industry technologies/tools in classroom setting). Proposals are encouraged to elaborate on how the proposed work is anticipated to impact student education on campus and/or the broader academic community.

Intel Note:

As an industry leader, Intel pushes the boundaries of technology to make amazing experiences possible for every person on earth. From powering the latest devices and the cloud you depend on to driving policy, diversity, sustainability, and education, we create value for our stockholders, customers, and society. Intel expects suppliers in our supply chain to be strong partners in making Intel successful through support of Intel's goals and commitments to diversity, sustainability, and education.

In light of Intel's strong commitment to diversity and creating an inclusive environment, in your proposal please address: (a) your organization's commitment to diversity and inclusion with respect to race, national origin, gender, veterans, individuals with diverse abilities and LGBTQ, (b) a summary of your performance in this area and any initiatives you are pursuing, and (c) the diverse team you propose for this project, including leadership, support, and any subcontracting you propose (such as to minority- or women-owned businesses).

Intellectual Property

This solicitation affords proposers the option of submitting proposals for the award of a grant or gift, a sponsored research agreement, or other agreement as appropriate. Intel reserves the right to negotiate the final choice of agreement.

The final award terms are expected to follow one or the other of two high-level intellectual property (IP) approaches. Either: (1) Intel and the university will jointly agree that IP developed under a grant or gift will be placed in the public domain, including offering software under an open-source license, or (2) Intel and the university will negotiate a sponsored research agreement with more specific IP terms, which, at a minimum, will require the university to grant Intel and other sponsors (if any) a broad non-exclusive royalty free license to foreground IP.

It is a requirement to follow approach (1) if a project's software development directly enhances the Lava software framework or builds on pre-existing INRC shared software libraries.

Please note that Intel is unable to receive proposals under an obligation of confidentiality. All proposals submitted should therefore include only public information. Accepted proposals may be published to the INRC member site for community reference (*i.e.* visible to all other members engaged in INRC research), specifically sections 1-7. Groups will have control over all such content on the INRC website and may request for their project details not to be shared at all in this manner with other members, if so desired.

Point of Contact for Inquiries and Submissions

Proposal submissions and related inquiries should be sent to INRC_Project_Proposals@intel.com.

References

[1] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. Fonseca Guerra, P. Joshi, P. Plank and S. Risbud, "Advancing Neuromorphic Computing with Loihi: A Survey of Results and Outlook," *Proceedings of the IEEE,* pp. 911-934, 2021.

[2] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang and H. Wang, "Loihi: a Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro,* 2018.

[3] L. Buesing, J. Bill, B. Nessler and W. Maass, "Neural Dynamics as Sampling: A Model for Stochastic Computation in Recurrent Networks of Spiking Neurons," *PLOS Computational Biology,* vol. 7, no. 11, 2011.

[4] D. Kleyko, M. Davies, E. P. Frady, P. Kanerva, S. J. Kent, B. A. Olshausen, E. Osipov, J. M. Rabaey, D. A. Rachkovskij, A. Rahimi and F. T. Sommer, "Vector Symbolic Architectures as a Computing Framework for Nanoscale Hardware," *arXiv 2106.05268,* 2021.

[5] Y. Sandamirskaya, "Dynamic neural fields as a step toward cognitive neuromorphic architectures," *Frontiers in Neuroscience,* vol. 7, no. 1662-453X, p. 276, 2014.

[6] D. M. Paiton, S. Shepard, K. H. R. Chan and B. A. Olshausen, "Subspace Locally Competitive Algorithms," in *NICE '20*, Heidelberg, Germany, 2020.

[7] Q. Li and C. Pehlevan, "Minimax Dynamics of Optimally Balanced Spiking Networks of Excitatory and Inhibitory Neurons," *arXiv 2006.08115,* 2021.

[8] S. Ludwig, J. Hartjes, B. Pol, G. Rivas and J. Kwisthout, "A spiking neuron implementation of genetic algorithms for optimization," in *BNAIC/Benelearn Communications in Computer and Information Science*, Leiden, 2020.

[9] D. Pecevski, L. Buesing and W. Maass, "Probabilistic Inference in General Graphical Models through Sampling in Stochastic Networks of Spiking Neurons," *PLOS Computational Biology,* vol. 7, pp. 1-25, 2011.

[10] S. Habenschuss, Z. Jonke and W. Maass, "Stochastic Computations in Cortical Microcircuit Models," *PLOS Computational Biology,* vol. 9, pp. 1-28, 2013.

[11] H. Jang, O. Simeone, B. Gardner and G. Andre, "An Introduction to Probabilistic Spiking Neural Networks: Probabilistic Models, Learning Rules, and Applications,," *IEEE Signal Processing Magazine,* vol. 36, no. 6, pp. 64-77, November 2019.

[12] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Communications,* 2020.

[13] F. Zenke and E. O. Neftci, "Brain-inspired learning on neuromorphic substrates," *Proceedings of the IEEE,* vol. 109, no. 5, pp. 935-950, 2021.

[14] J. Sacramento, R. P. Costa, Y. Bengio and W. Senn, "Dendritic error backpropagation in deep cortical microcircuits," *arXiv:1801.00062,* Dec 2017.

[15] B. Scellier and Y. Bengio, "Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation," *Frontiers in Computational Neuroscience,* vol. 11, no. 1662-5188, p. 24, 2017.

[16] K. Stewart, G. Orchard, S. B. Shrestha and E. Neftci, "Online Few-Shot Gesture Learning on a Neuromorphic Processor," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 10, no. 4, pp. 512-521, 2020.

[17] B. Illing, J. Ventura, G. Bellec and W. Gerstner, "Local plasticity rules can learn deep representations using self-supervised contrastive predictions," *arXiv 2010.08262,* 2021.

[18] A. van den Oord, Y. Li and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv 1807.03748,* 2019.

[19] X. Chen, H. Fan, R. Girshick and K. He, "Improved Baselines with Momentum Contrastive Learning," *arXiv 2003.04297,* 2020.

[20] G. Parisi, "A nenory which forgets," *J Phys A,* vol. 10, no. 19, p. 617, 1986.

[21] E. P. Frady, D. Kleyko and F. T. Sommer, "Variable Binding for Sparse Distributed Representations: Theory and Applications," *arXiv preprint arXiv:2009.06734,* 2020.

[22] S. Fusi, P. J. Drew and L. F. Abbott, "Cascade models of synaptically stored memories.," *Neuron,* vol. 45, no. 4, pp. 599-611, 2005.

[23] S. Romani, D. J. Amit and Y. Amit, "Optimizing one-shot learning with binary synapses," *Neural Computation,* vol. 20, no. 8, pp. 1928-1950, 2008.

[24] S. . Fusi and L. F. Abbott, "Limits on the memory storage capacity of bounded synapses," *Nature Neuroscience,* vol. 10, no. 4, pp. 485-493, 2007.

[25] D. D. Ben Dayan Rubin and S. Fusi, "Long memory lifetimes require complex synapses and limited sparseness," *Frontiers in Computational Neuroscience,* vol. 1, p. 7, 2007.

[26] M. K. Benna and S. . Fusi, "Computational principles of synaptic memory consolidation," *Nature Neuroscience,* vol. 19, no. 12, pp. 1697-1706, 2016.

[27] A. . Roxin and S. . Fusi, "Efficient Partitioning of Memory Systems and Its Importance for Memory Consolidation," *PLOS Computational Biology,* vol. 9, no. 7, pp. 1-13, 2013.

[28] N. . Chenkov, H. . Sprekeler and R. . Kempter, "Memory replay in balanced recurrent networks," *PLOS Computational Biology,* vol. 13, no. 1, p. , 2017.

[29] M. . Katkov, S. . Romani, M. . Tsodyks and M. . Tsodyks, "Memory Retrieval from First Principles.," *Neuron,* vol. 94, no. 5, pp. 1027-1032, 2017.

[30] B. Ravindran and A. G. Barto, "Relativized options:

choosing the right transformation," 2003. [Online]. Available: https://aaai.org/papers/icml/2003/icml03-080.pdf. [Accessed 17 8 2021].

[31] S. M. Andrew G. Barto, "Recent Advances in Hierarchical Reinforcement Learning," *Discrete Event Dynamic Systems,* vol. 13, p. 341–379, 2003.

[32] M. . Botvinick, Y. . Niv and A. C. Barto, "Hierarchically Organized Behavior and Its Neural Foundations: A Reinforcement Learning Perspective.," *Cognition,* vol. 113, no. 3, pp. 262-280, 2009.

[33] Y. . Niv, R. . Daniel, A. . Geana, S. J. Gershman, Y. C. Leong, A. . Radulescu and R. C. Wilson, "Reinforcement Learning in Multidimensional Environments Relies on Attention Mechanisms," *The Journal of Neuroscience,* vol. 35, no. 21, pp. 8145-8157, 2015.

[34] J. Zhang, H. Yu and W. Xu, "Hierarchical Reinforcement Learning By Discovering Intrinsic Options," in *international conference on Learning Representations*, 2021.

[35] N. Imam and T. A. Cleland, "Rapid learning and robust recall in a neuromorphic olfactory circuit," *(in review).*

[36] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal and Y. Bengio, "Toward Causal Representation Learning," *IEEE Advances in Machine Learning and Deep Neural Networks,* vol. 109, no. 5, 2021.

[37] Z. Jonke, S. Habenschuss and W. Maass, "Solving Constraint Satisfaction Problems with Networks of Spiking Neurons," *Frontiers in Neuroscience,* vol. 10, p. 118, 2016.

[38] E. P. Frady, S. J. Kent, B. A. Olshausen and F. T. Sommer, "Resonator Networks, 1: An Efficient Solution for Factoring High-Dimensional, Distributed Representations of Data Structures," *Neural Computation,* vol. 32, no. 12, pp. 2311-2331, 2020.

[39] G. Tang, A. Shah and K. P. Michmizos, "Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM," in *IROS*, 2019.

[40] R. Kreiser, A. Renner, V. R. C. Leite, B. Serhan, C. Bartolozzi, A. Glover and Y. Sandamirskaya, "An On-chip Spiking Neural Network for Estimation of the Head Pose of the iCub Robot," *Frontiers in Neuroscience,* vol. 14, no. 1662-453X, p. 551, 2020.

[41] C. A. R. Hoare, "Communicating sequential processes," *Communications of the ACM,* vol. 21, no. 8, pp. 666-677, 1978.

[42] M. Lohstroh, "Reactors: A Deterministic Model of Concurrent Computation for Reactive Systems," EECS Department, University of California, Berkeley, 2020.

[43] G. Susi, P. Garcés, E. Paracone, A. Cristini, M. Salerno, F. Maestú and E. Pereda, "FNS allows efficient event-driven spiking neural network simulations based on a neuron model supporting spike latency," *Scientific Reports,* vol. 11, no. 1, p. 12160, 2021.

[44] S. B. Shrestha and G. Orchard, "SLAYER: Spike Layer Error Reassignment in Time," in *32nd Conference on Neural Information Processing Systems*, Montréal, Canada, 2018.

[45] C. Pehlevan, "A Spiking Neural Network with Local Learning Rules Derived From Nonnegative Similarity Matching," *arXiv,* 2019.

Legal Disclaimers

The issuance of this RFP and the submission of a response by a respondent or the acceptance of such a response by Intel Corporation ("Intel") does not obligate Intel in any manner. The RFP is not an offer or a contract. Intel is not obligated to contract for any of the products/services described in the RFP. Intel reserves the right to:

1) amend, modify or withdraw this RFP;
2) revise any requirement of this RFP;
3) waive any requirements of this RFP that are not material;
4) seek clarifications and revisions of responses to this RFP;
5) require supplemental statements or information from any responsible party;
6) accept or reject any or all responses to this RFP;
7) extend the deadline for submission of responses to this RFP or otherwise modify the schedule set forth in this RFP;
8) negotiate potential terms with any respondent to this RFP;
9) engage in discussions with any respondent to this RFP to correct and/or clarify responses;
10) require clarification at any time during the procurement process and/or require correction of responses for the purpose of assuring a full and complete understanding of a respondent's proposal and/or determine a respondent's compliance with the requirements of the solicitation; and
11) cancel, or reissue in whole or in part, this RFP, if Intel determines in its sole discretion that it is its best interest to do so.

Intel may exercise the foregoing rights at any time without notice and without liability to any respondent or any other party for its expenses incurred in preparation of responses hereto or otherwise. All costs associated with responding to this RFP will be at the sole cost and expense of the respondent. Intel makes no representation or warranty and shall incur no liability under any law, statute, rules or regulations as to the accuracy, reliability or completeness of this RFP.

Appendix 1: Neuromorphic Algorithm Assessment

The purpose of this assessment is to determine (a) if the proposed neuromorphic algorithm is sufficiently well defined with properties that match well to a neuromorphic hardware implementation, (b) the general usefulness of the proposed algorithm, (c) if current Loihi silicon and software can support the algorithm, (d) how the algorithm will be assessed relative to state-of-the-art alternatives.

Algorithm definition and requirements

WHAT	What computational problem does the proposed algorithm solve?	*Ideally, this is a clear mathematical objective.*
	What learning paradigms are involved, if any?	*E.g. Supervised (online or offline?), self-supervised, unsupervised, reinforcement-based, associative, gradient-based adaptation, continual, etc.*
	What are the algorithm's essential "neuromorphic" properties?	*E.g. Temporal neuron models, binary activations, sparse spike/event-based communication, sparse connectivity, recurrence, E/I balance, parameter plasticity, structural plasticity, fully local synaptic & neural information processing, distributed data representations.*
	Are data input/output interfaces and encodings well defined?	*Does the algorithm operate on conventional data types, or spiking/event-based data types? Does it process time series data streams with temporal structure (e.g. video), or isolated, uncorrelated samples (e.g. images)?*
	How far towards a deployable neuromorphic solution will the proposed algorithmic research be taken?	*Is the aim to develop software that executes the algorithm on Loihi to process real-world data? Or is the goal a simulation-based demonstration believed to be compatible with Loihi HW? Something in between?*
WHY	How broadly applicable is the proposed algorithm?	*What real-world capabilities, applications, and technologies could this algorithm be used in?*
	Characterize the difficulty of the problem to be solved.	*E.g. NP-complete/hard, existing state-of-the-art DNN network scale, typical Energy-Delay-Product application constraints, typical CPU runtime, etc.*
	What value does a neuromorphic solution promise?	*Does the algorithm primarily improve the energy, speed, or data efficiency of existing algorithms? Or is there no known conventional solution?*
	Is the algorithm modular and composable?	*Can the algorithm be integrated into a larger application where the whole is greater than the sum of its parts?*
REQUIREMENTS	How mature is the proposed algorithmic approach?	*Are the key algorithmic or implementation ingredients understood or is it part of this proposal to develop such an understanding? What are the key open questions and risks? Do conventional ANN implementations exist?*
	At what problem scale is the algorithm expected to be demonstrated?	*What scale will be demonstrated relative to the scale demanded by useful and impactful real-world applications? (In terms of physical metrics like I/O dimensionality, #parameters, #neurons, dataset size, stored patterns/classes, etc.)*
	What requirements does the algorithm impose on Lava SW infrastructure or other algorithms?	*What infrastructure is expected to exist (or when will it need to exist)? Are there critical requirements on input/output bandwidth?*

	What can be said about the specific features and numeric precision required of the neuromorphic hardware?	*Note: Loihi's synaptic variables provide up to 1B, while neural variables offer 1B, 2B or 3B of (un)signed integer precision.*
	Does the algorithm require on-chip synaptic plasticity?	*If so, are the learning rules supported by Loihi's micro-code programmable learning engine, if known?*
	Does the algorithm currently depend on features not supported by Loihi 2?	*(If known) Examples: division or transcendental functions applied to neuron/synaptic state changes, non-local weight normalization or transport.*

Proposed approach and evaluation methodology

HOW	What methodology is being followed to develop this algorithm?	*Is the proposed algorithm from the category of mathematically derived algorithms or directly inspired by neuroscience modeling?*
	What software tools will be used to develop the algorithm?	*E.g. Lava, Brian, Nengo, Matlab, directly coded Python/C++, TensorFlow, PyTorch, SLAYER, SpyTorch, Fugu, etc.*
	How is the algorithm or initial neural network configured, parameterized, or trained?	*Does it rely on pre-training with back-propagation, manual parameter tuning, evolutionary methods or is the network configuration computed analytically?*
	Does the algorithm involve continual online learning? If so, how?	*Is new knowledge absorbed into existing resources or reliant on allocating new memory resources over time? How is forgetting mitigated? Does learning rely on assumptions of IID input data?*
	What neuron model(s) will be used?	*E.g. LIF, ALIF, CUBA, COBA, ReLU, Resonate-and-fire, Izhikevich, GLM, etc.*
	How is data coded in the network?	*E.g. spike-based or integer-valued events, temporal coding, rate coding, population coding, mixtures thereof.*
COMPETITION	How will the performance of the algorithm be evaluated?	*What are critical performance metrics? Will standardized benchmarks, datasets, simulation environments, etc. be used? If so, please list them. If not, how will performance be quantified in a replicatable manner?*
	What competitive state-of-the-art *conventional* and *neuromorphic* solutions exist today, if any, and what HW platforms do they run on?	*Which of the current solutions are state-of-the-art with respect to the metrics/benchmarks above? Does the proposer have access to a working reference implementation of any of these competitive algorithms?*
	What are the limitations or major pain points of current solutions?	*In what way are current solutions inadequate? Are they failing to meet real-world application needs in terms of key metrics?*

Note: A neuromorphic algorithm solves a new, specific and well-defined computational problem by exploiting features of the underlying neuromorphic hardware system following established mathematical principles in machine learning, optimization, etc. or might be inspired by neuroscientific, bio-inspired modeling. As an algorithmic project, there might still be open theoretical questions or how this algorithm can be supported by neuromorphic hardware, yet the value for overcoming those risks should be clearly spelled out and significant. If the theoretical open questions are too broad, the research best belongs in the Theory vector (RV1).

In contrast, applications mostly build on top of previously coded algorithms that each have little execution risk. Application projects draw their novelty and impact from the composition of algorithms deployed and the value of real-world problem solved.

Appendix 2: Neuromorphic Application Assessment

The purpose of this assessment is to determine (a) if the proposed application is sufficiently well defined, (b) if the value that neuromorphic technology can provide is well understood and compelling, (c) if the practical requirements of the application can be satisfied with algorithms, software, and systems available today, and (d) the likelihood of demonstrating an unambiguous state-of-the-art solution.

Application definition and requirements

WHAT	What is the task?	*Please provide a concise high-level description of the application.* *Are there any references to learn more about the task?*
	What are the key computationally hard components of the task?	*Examples might be DNNs, ML algorithms, standard optimization objectives (LASSO, QUBO, MLIP, graph search, etc.) If non-standard, please provide a mathematical description or references to computational problem. What proportion of the task's hard computational components will be solved neuromorphically?*
WHY	What is the impact of solving the task?	*Articulate in terms of value to the end user/customer. Why will anyone care that this task is successfully accomplished?*
	Is there already an existing market or a path to commercialization for this application?	*Who are the customers? What is the path to commercialization? Any obstacles or recent enabling developments? What is the size of an existing market?*
	How generalizable or broadly applicable is a solution to this task?	*How easy is it to generalize the task to other domains?* *What else could a superior implementation to components of the task be applied to?*
REQUIREMENTS	What are the key metrics to assess the performance or quality of this task?	*Examples could be energy or time to solution, accuracy or result with respect to some reference, area, cost.*
	Are there critical data IO, throughput, latency, closed-loop requirements?	
	Are there requirements on setup time?	*Setup time may include compiling a program or model, loading it onto a HW platform?* *If the application is typically launched once and executed for a long time, then the answer is likely no. If otherwise, please explain.*
	What degree of programmability or flexibility is required by the HW compute platform?	*Does the task require the flexibility of a general-purpose CPU to maximize developer productivity or satisfy other constraints or is the task best served by a custom ASIC or anything in-between?*
	Does the task require any real-time adaptation?	*Is the program or model entirely pre-configured or pre-trained before deployment or does it have to adapt after deployment by itself based on data?* *What type of adaptation is required? Backpropagation or other learning rules?*

Proposed implementation and state-of-the-art baseline

HOW	What neuromorphic methods will be used to solve the task?	*What algorithms, software, and other critical neuromorphic ingredients will be used? Please provide references wherever possible.*
	How mature are these methods; what exists today versus what needs to be invented/proven?	*Have the necessary algorithms been modeled successfully to date? Have they been mapped to neuromorphic hardware and shown to work at the scale demanded by the application? Have all application precision requirements been considered? Do the relevant Lava software modules exist today? Are the modules interoperable/composable?*
COMPETITION	What competitive state-of-the-art solutions exist today, if any?	*What are critical performance metrics? In what way are current solutions state of the art with respect to all or some metrics?* *Does the proposer have access to a working reference implementation to the problem based on conventional systems?*
	What are the limitations or major pain points of current solutions?	*How and why are they insufficient? In what way do current solutions not address the desired key metric requirements?*
	How will the neuromorphic solution be evaluated against other solutions?	Are there standardized benchmarks to evaluate the proposed solution versus state-of-the-art alternatives? *If so, please describe and provide references if possible. If not, how will success be defined?*