# Energy efficient synaptic plasticity

**Ho Ling Li[1] and Mark C. W. van Rossum[1,2]***

**\*For correspondence:**
mark.vanrossum@nottingham.ac.uk

[1]School of Psychology; [2]School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, U.K.

**Abstract** Many aspects of the brain's design can be understood as the result of evolutionary drive towards metabolic efficiency. In addition to the energetic costs of neural computation and transmission, experimental evidence indicates that synaptic plasticity is metabolically demanding as well. As synaptic plasticity is crucial for learning, we examine how these metabolic costs enter in learning. We find that when synaptic plasticity rules are naively implemented, training neural networks requires extremely large amounts of energy when storing many patterns. We propose that this is avoided by precisely balancing labile forms of synaptic plasticity with more stable forms. This algorithm, termed synaptic caching, boosts energy efficiency manifold and can be used with any plasticity rule, including back-propagation. Our results yield a novel interpretation of the multiple forms of neural synaptic plasticity observed experimentally, including synaptic tagging and capture phenomena. Furthermore our results are relevant for energy efficient neuromorphic designs.

## Introduction

The human brain only weighs 2% of the total body mass, but is responsible for 20% of resting metabolism (*Attwell and Laughlin, 2001*; *Harris et al., 2012*). The brain's energy need is believed to have shaped many aspects of its design, such as its sparse coding strategy (*Levy and Baxter, 1996*; *Lennie, 2003*), the biophysics of the mammalian action potential (*Alle et al., 2009*; *Fohlmeister, 2009*), and synaptic failure (*Levy and Baxter, 2002*; *Harris et al., 2012*). As the connections in the brain are adaptive, one can design synaptic plasticity rules that further reduce the energy required for information transmission, for instance by sparsifying connectivity (*Sacramento et al., 2015*). But in addition to the costs associated to neural information processing, experimental evidence suggests that memory formation, presumably corresponding to synaptic plasticity, is itself an energetically expensive process as well (*Mery and Kawecki, 2005*; *Plaçais and Preat, 2013*; *Jaumann et al., 2013*; *Plaçais et al., 2017*).

To estimate the amount of energy required for plasticity, *Mery and Kawecki* (*2005*) subjected fruit flies to associative conditioning spaced out in time, resulting in long-term memory formation. After training, the fly's food supply was cut off. Flies exposed to the conditioning died some 20% quicker than control flies, presumably due to the metabolic cost of plasticity. Likewise, fruit flies doubled their sucrose consumption during the formation of aversive long-term memory (*Plaçais et al., 2017*), while forcing starving fruit flies to form such memories reduced lifespan by 30% (*Plaçais and Preat, 2013*). A massed learning protocol, where pairings are presented rapidly after one another, leads to less permanent forms of learning that don't require protein synthesis. Notably this form of learning is energetically less costly (*Mery and Kawecki, 2005*; *Plaçais and Preat, 2013*). In rats (*Gold, 1986*) and humans (*Hall et al., 1989*, but see *Azari, 1991*) beneficial effects of glucose on memory have been reported, although the intricate regulation of energy complicates interpretation of such experiments (*Craft et al., 1994*).

Motivated by the experimental results, we analyze the metabolic energy required to form
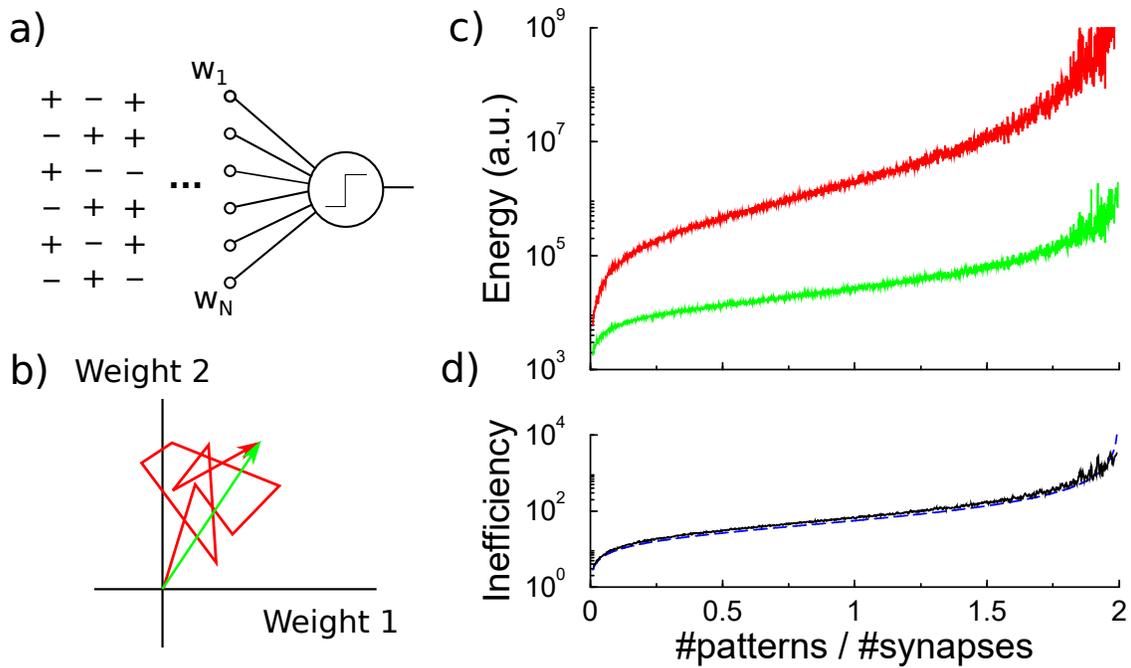
**Figure 1. Energy efficiency of perceptron learning.** (a) A perceptron cycles through the patterns and updates its synaptic weights until all patterns produce their correct target output. (b) During learning the synaptic weights follow approximately a random walk (red path) until they find the solution (grey region). The energy consumed by the learning corresponds to the total length of the path (under the $L_1$ norm). (c) The energy required to train the perceptron diverges when storing many patterns (red curve). The minimal energy required to reach the correct weight configuration is shown for comparison (green curve). (d) The inefficiency, defined as the ratio between actual and minimal energy plotted in panel c, diverges as well (black curve). The overlapping blue curve corresponds to the theory, Eq. 3 in the text.

**Figure 1–Figure supplement 1.** Energy inefficiency as a function of exponent $\alpha$ in the energy function.

---

44  associative memories in neuronal networks. We demonstrate that traditional learning algorithms
45  are metabolically highly inefficient. Therefore we introduce a synaptic caching algorithm that is
46  consistent with synaptic consolidation experiments, and distributes learning over transient and
47  persistent synaptic changes. This algorithm increases efficiency manifold. Synaptic caching yields a
48  novel interpretation to various aspects of synaptic physiology, and suggests more energy efficient
49  neuromorphic designs.

50  ## Results

51  ### Inefficiency of perceptron learning

52  To examine the metabolic energy cost associated to synaptic plasticity, we first study the perceptron.
53  A perceptron is a single artificial neuron that attempts to binary classify input patterns. It forms the
54  core of many artificial networks and has been used to model plasticity in cerebellar Purkinje cells.
55  We consider the common case where the input patterns are random patterns each associated to a
56  randomly chosen binary output. Upon presentation of a pattern, the perceptron output is calculated
57  and compared to the desired output. The synaptic weights are modified according to the perceptron
58  learning rule, Figure 1A. This is repeated until all patterns are classified correctly (*Rosenblatt, 1962*,
59  see Methods and Materials). Typically, the learning takes multiple iterations over the whole dataset
60  ('epochs').
61  As it is not well known how much metabolic energy is required to modify a biological synapse, and
62  how this depends on the amount of change and the sign of the change, we propose a parsimonious
63  model. We assume that the metabolic energy for every modification of a synaptic weight is
64  proportional to the amount of change, no matter if this is positive or negative. The total metabolic

65  cost $M$ (in arbitrary units) to train a perceptron is the sum over the weight changes of synapses

$$M_{\text{perc}} = \sum_{i=1}^{N} \sum_{t=1}^{T} |w_i(t) - w_i(t-1)|^{\alpha}, \tag{1}$$

66  where $N$ is the number of synapses, $w_i$ denotes the synaptic weight at synapse $i$, and $T$ is the total
67  number of time-steps required to learn the classification. The exponent $\alpha$ is set to one, but our
68  results below are similar whenever $0 \leq \alpha \lesssim 2$, Figure 1-Figure supplement 1. As there is evidence that
69  synaptic depression involves different pathways than synaptic potentiation (e.g. *Hafner et al., 2019*),
70  we also tried a variant of the cost function where only potentiation costs energy and depression
71  does not. This does not change our results, Figure 1-Figure supplement 1.

72  Learning can be understood as a search in the space of synaptic weights for a weight vector
73  that leads to correct classification of all patterns, Figure 1B. The synaptic weights approximately
74  follow a random walk (Methods and Materials), and the metabolic cost is proportional to the length
75  of this walk under the $L_1$ norm, Eq. 1. The perceptron learning rule is energy inefficient, because
76  repeatedly, weight modifications made to correctly classify one pattern are partly undone when
77  learning another pattern. However, as both processes require energy, this is inefficient.

78  The energy required by the perceptron learning rule depends on the number of patterns $P$
79  to be classified. The set of correct weights spans a cone in $N$-dimensional space (grey region in
80  Figure 1B). As the number of patterns to be classified increases, the cone containing correct weights
81  shrinks and the random walk becomes longer (*Gardner, 1987*). Near the critical capacity of the
82  perceptron ($P = 2N$), the number of epochs required diverges as $(2 - P/N)^{-2}$, *Opper* (*1988*). The
83  energy required, which is proportional to the number of updates that the weights undergo, follows
84  a similar behavior, Figure 1C.

85  It is useful to consider the theoretical minimal energy required to classify all patterns. The most
86  energy efficient algorithm would somehow directly set the synaptic weights to their desired final
87  values. Geometrically, the random walk trajectory of the synaptic weights to the target is replaced
88  by a path straight to the correct weights (green arrow in Figure 1B). Given the initial weights $w_i(0)$
89  and the final weights $w_i(T)$, the energy required in this idealized case is

$$M_{\text{min}} = \sum_{i} |w_i(T) - w_i(0)|. \tag{2}$$

90  While the minimal energy also grows with memory load (Methods and Materials), it increases less
91  steeply, Figure 1C.

92  We express the metabolic efficiency of a learning algorithm as the ratio between the energy the
93  algorithm requires and the minimal energy (the gap between the two log-scale curves in Figure 1C).
94  As the number of patterns increases, the inefficiency of the perceptron rule rapidly grows as (see
95  Methods and Materials)

$$\frac{M_{\text{perc}}}{M_{\text{min}}} = \frac{\sqrt{\pi P}}{2 - P/N}, \tag{3}$$

96  which fits the simulations very well, Figure 1D, black curve and dashed blue curve.

97  There is evidence that both cerebellar and cortical neurons are operating close to their maximal
98  memory capacity (*Brunel et al., 2004*; *Brunel, 2016*). Indeed, it would appear wasteful if this were
99  not the case. However, the above result demonstrates that for instance classifying 1900 patterns by
100  a neuron with 1000 synapses with the traditional perceptron learning requires about ~900 times
101  more energy than minimally required. As the fruit-fly experiments indicate that even storing a
102  single association in long-term memory is already metabolically expensive, storing many memories
103  would thus require very large amounts of energy if the biology would naively implement these
104  learning rules.

## Synaptic caching

106  How can the conflicting demands of energy efficiency and high storage capacity be met? The
107  minimal energy argument presented above suggests a way to increase energy efficiency. There
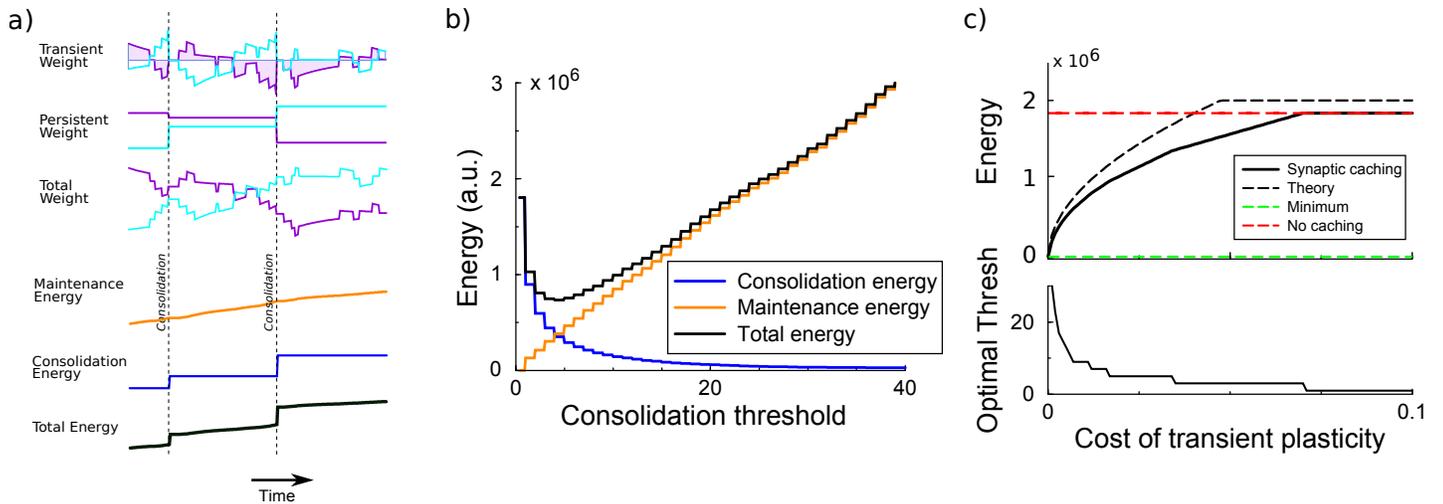
**Figure 2. Synaptic caching algorithm.** (a) Changes in the synaptic weights are initially stored in metabolically cheaper transient decaying weights. Here two example weight traces are shown (blue and magenta). The total synaptic weight is composed of transient and persistent forms. Whenever any of the transient weights exceed the consolidation threshold, the weights become persistent and the transient values are reset (vertical dashed line). The corresponding energy consumed during the learning process consists of two terms: the energy cost of maintenance is assumed to be proportional to the magnitude of the transient weight (shaded area in top traces); energy cost for consolidation is incurred at consolidation events. (b) The total energy is composed of the energy to occasionally consolidate and the energy to support transient plasticity. Here it is minimal for an intermediate consolidation threshold. (c) The amount of energy required for learning with synaptic caching, in the absence of decay of the transient weights (black curve). When there is no decay and no maintenance cost the energy equals the minimal one (green line) and the efficiency gain is maximal. As the maintenance cost increases, the optimal consolidation threshold decreases (lower panel) and the total energy required increases, until no efficiency is gained at all by synaptic caching.

**Figure 2–Figure supplement 1.** Synaptic caching in a spiking neuron with a biologically plausible perceptron-like learning rule.

---

108  are forms of plasticity - anaesthesia resistant memory in flies and early-LTP/LTD in mammals - that
109  decay and do not require protein synthesis. Such transient synaptic changes can be induced using
110  a massed, instead of a spaced, stimulus presentation protocol. Fruit-fly experiments show that
111  this form of plasticity is much less energy-demanding than long-term memory (*Mery and Kawecki,*
112  *2005*; *Plaçais and Preat, 2013*; *Plaçais et al., 2017*). In mammals there is evidence that synaptic
113  consolidation, but not transient plasticity, is suppressed under low energy conditions (*Potter et al.,*
114  *2010*). Inspired by these findings we propose that the transient form of plasticity constitutes a
115  synaptic variable that accumulates the synaptic changes across multiple updates in a less expensive
116  transient form of memory; only occasionally the changes are consolidated. We call this *synaptic*
117  *caching.*
118      Specifically, we assume that each synapse is comprised of a transient component $s_i$ and a
119  persistent component $l_i$. The total synaptic weight is their sum, $w_i = s_i + l_i$. We implement synaptic
120  caching as follows, Figure 2A: For every presented pattern, changes in the synaptic strength are
121  calculated according to the perceptron rule and are accumulated in the transient component that
122  decays exponentially to zero. If, however, the absolute value of the transient component of a
123  synapse exceeds a certain consolidation threshold, all synapses of that neuron are consolidated
124  (vertical dashed line in Figure 2A), the value of the transient component is added to the persistent
125  weight, and the transient weight is reset to zero.
126      The efficiency gain of synaptic caching depends on the limitations of transient plasticity. If the
127  transient synaptic component could store information indefinitely at no metabolic cost, consolida-
128  tion could be postponed until the end of learning and the energy would equal the minimal energy
129  Eq. 2. Hence the efficiency gain would be maximal. However, we assume that the efficiency gain of
130  synaptic caching is limited because of two effects: 1) The transient component decays exponentially
131  (with a time-constant $\tau$). 2) There might be a maintenance cost associated to maintaining the tran-
132  sient component. Biophysically, transient plasticity might correspond to an increased/decreased

133 vesicle release rate (**_Padamsey and Emptage, 2014_**; **_Costa et al., 2015_**) so that it diverges from its
134 optimal value (**_Levy and Baxter, 2002_**).

135     To estimate the energy saved by synaptic caching, we assume that the maintenance cost is
136 proportional to the transient weight itself and incurred every time-step $\Delta t$ (shaded area in the top
137 traces of Figure 2A)

$$M_{\text{trans}} = c \sum_i \sum_t |s_i(t)|.$$

138 While experiments indicate that transient plasticity is metabolically far less demanding than the
139 persistent form, the precise value of the maintenance cost is not known. We encode it in the
140 constant $c$; the theory also includes the case that $c$ is zero. It is straightforward to include a cost
141 term for changing the transient weight (Methods); such a cost would reduce the efficiency gain
142 attainable by synaptic caching.

143     Next we need to include the energetic cost of consolidation. Currently it is unknown how
144 different components of synaptic consolidation, such as signaling, protein synthesis, transport to
145 the synapses and changing the synapse, contribute to this cost. We assume the metabolic cost
146 to consolidate the synaptic weights is $M_{\text{cons}} = \sum_i \sum_t |l_i(t) - l_i(t-1)|$. This is identical to Eq. 1, but in
147 contrast to standard perceptron learning where synapses are consolidated every time a weight
148 is updated, now changes in the persistent component $l_i$ only occur when consolidation occurs.
149 One could add a maintenance cost term to the persistent weight as well, in that case postponing
150 consolidation would save even more energy.

### Efficiency gain from synaptic caching

152 To maximize the efficiency gain achieved by synaptic caching one needs to tune the consolidation
153 threshold, Figure 2B. When the threshold is low, consolidation occurs often and the energy ap-
154 proaches the one without synaptic caching. When on the other hand the consolidation threshold
155 is high, the expensive consolidation process occurs rarely, but the maintenance cost of transient
156 plasticity is high, moreover the decay will lead to forgetting of unconsolidated memories, slowing
157 down learning and increasing the energy cost. Thus the consolidation energy decreases for larger
158 thresholds, whereas the maintenance energy increases, Figure 2B (see Methods and Materials). As
159 a result of this trade-off there is an optimal threshold, which depends on the decay and the mainte-
160 nance cost, that balances persistent and transient forms of plasticity. To analyze the efficiency gain
161 below we numerically optimize the consolidation threshold.

162     First we consider the case when the transient component does not decay. Figure 2C shows the
163 energy required to train the perceptron. When the maintenance cost is absent ($c = 0$), consolidation
164 is best postponed until the end of the learning and the energy is as low as the theoretical minimal
165 bound. As $c$ increases, it becomes beneficial to consolidate more often, i.e. the optimal threshold
166 decreases, Figure 2C bottom panel. The required energy increases until the maintenance cost
167 becomes so high that it is better to consolidate after every update, the transient weights are not
168 used, and no energy is saved with synaptic caching. The efficiency is well estimated by analysis
169 presented in the Methods and Materials, Figure 2C (theory).

170     Next, we consider what happens when the transient plasticity decays. We examine the energy
171 and learning time as a function of the decay rate for various levels of maintenance cost, Figure 3.
172 As stated above, if there is no decay, efficiency gain can be very high; the consolidation threshold
173 has no impact on the learning time, Figure 3 bottom. In the other limit, when the decay is rapid
174 (right-most region), it is best to consolidate frequently as otherwise information is lost. As expected,
175 the metabolic cost is high in this case.

176     The regime of intermediate decay is quite interesting. When maintenance cost is high, it is of
177 primary importance to keep learning time short, and in fact the learning time can be lower than in
178 a perceptron without decay, Figure 3, bottom, light curves. When on the other hand maintenance
179 cost is low, the optimal solution is to set the consolidation threshold high so as to minimize the
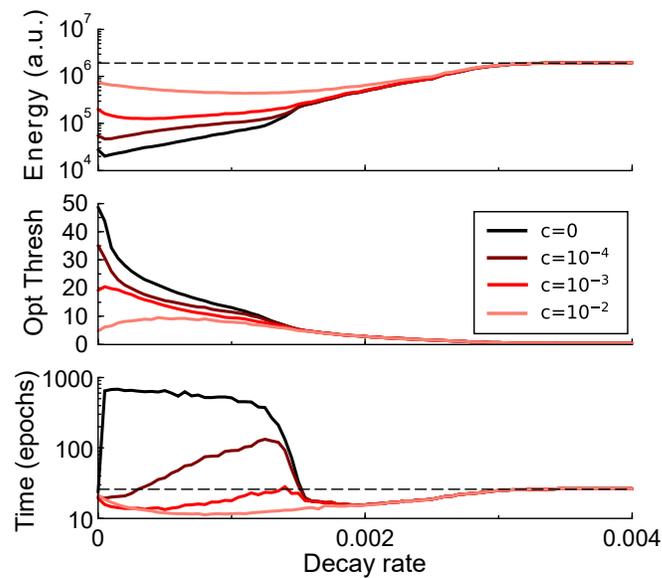
**Figure 3. Synaptic caching and decaying transient plasticity.** The amount of energy required, the optimal consolidation threshold, and the learning time as a function of the decay rate of transient plasticity for various values of the maintenance cost. Broadly, stronger decay will increase the energy required and hence reduce efficiency. With weak decay and small maintenance cost, the most energy-saving strategy is to accumulate as many changes in the transient forms as possible, thus increasing the learning time (darker curves). However, when maintenance cost is high, it is optimal to reduce the threshold and hence learning time. Dashed lines denote the results without synaptic caching.

**Figure 3–Figure supplement 1.** The effects of consolidation threshold on energy cost and learning time.

---

180 number of consolidation events, even if this means a longer learning time, Figure 3, bottom, dark
181 curves.

182 For intermediate decay rates, the consolidation threshold trades off between learning time and
183 energy efficiency, Figure 3–Figure supplement 1A. That is, by setting the consolidation threshold the
184 perceptron can learn either rapidly or efficiently. Such a trade-off could be of biological relevance.
185 We found a similar trade-off in multi-layer perceptrons (see below), Figure 3–Figure supplement 1B.
186 (although we found no evidence that learning can be sped up there).

187 In summary, when the transient component decays the learning dynamics is altered, and
188 synaptic caching can not only reduce metabolic cost but can also reduce learning time.

189 Next, to show that synaptic caching is a general principle, we implement synaptic caching
190 in a spiking neural network with a biologically plausible perceptron-like learning rule proposed
191 by *D'Souza et al.* (*2010*). The optimal scenario where the transient weights do not decay and have
192 no maintenance cost is assumed. The network is able to save 80% of the energy with synaptic
193 caching, Figure 2–Figure supplement 1. Hence, efficiency gains from synaptic caching do not rely on
194 exact implementation.

195 In the above implementation of synaptic caching, consolidation of all synapses was triggered
196 when transient plasticity at a single synapse exceeded a certain threshold. This resembles the
197 synaptic tagging and capture phenomenon where plasticity induction leads to transient changes
198 and sets a tag; only strong enough stimulation results in proteins being synthesized and being
199 delivered to all tagged synapses, consolidating the changes (*Frey and Morris, 1997*; *Barrett et al.,*
200 *2009*). There is a number of ways synapses could interact, Figure 4A. First, consolidation might be
201 set to occur whenever transient plasticity at a synapse crosses the threshold and only that synapse
202 is consolidated. Second, a hypothetical signal might send to the soma and consolidation of all
203 synapses occurs once transient plasticity at any synapse crosses the threshold (used in Figs. 2
204 and 5). Thirdly, a hypothetical signal might be accumulated in or near the soma and consolidation of
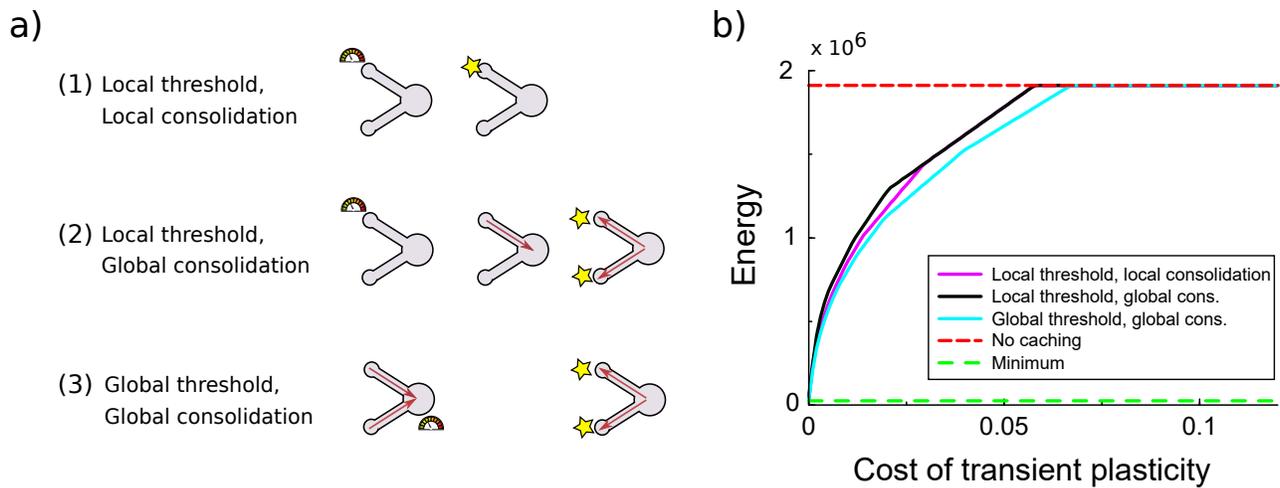
**Figure 4. Comparison of various variants of the synaptic caching algorithm.** (a) Schematic representation of variants to decide when consolidation occurs. From top to bottom: 1) Consolidation (indicated by the star) occurs whenever transient plasticity at a synapse crosses the consolidation threshold and only that synapse is consolidated. 2) Consolidation of all synapses occurs once transient plasticity at any synapse crosses the threshold. 3) Consolidation of all synapses occurs once the total transient plasticity across synapses crosses the threshold. (b) Energy required to teach the perceptron is comparable across algorithm variants. Consolidation thresholds were optimized for each algorithm and each maintenance cost of transient plasticity individually. In this simulation the transient plasticity did not decay.

---

205   all synapses occurs once this total transient plasticity across synapses crosses the threshold. Only
206   cases 2 and 3 are consistent with synaptic tagging and capture experiments, where consolidation of
207   one synapse also leads to consolidation of another synapse that would otherwise decay back to
208   baseline (*Frey and Morris, 1997*; *Sajikumar et al., 2005*). However, all variants lead to comparable
209   efficiency gains, Figure 4B.
210      In summary we see that synaptic caching can in principle achieve large efficiency gains, bringing
211   efficiency close to the theoretical minimum.

212 **Synaptic caching in multi-layer networks**
213   Since the perceptron is a rather restrictive framework, we wondered whether the efficiency gain of
214   synaptic caching can be transferred to multi-layer networks. Therefore we implement a multi-layer
215   network trained with back-propagation. Back-propagation networks learn the associations of
216   patterns by approaching the minimum of the error function through stochastic gradient descent.
217   We use a network with one hidden layer with by default 100 units to classify hand-written digits from
218   the MNIST dataset. As we train the network, we intermittently interrupt the learning to measure
219   the energy consumed for plasticity thus far and measure the performance on a held-out test-set.
220   This yields a curve relating energy to accuracy.
221      Similar to a perceptron, learning without synaptic caching is metabolically expensive in a back-
222   propagation network. Until reaching maximal accuracy, energy rises approximately exponentially
223   with accuracy, after which additional energy do not lead to further improvement. When the learning
224   rate is sufficiently small, the metabolic cost of plasticity is independent of the learning rate. At
225   larger learning rates, learning no longer converges and energy goes up steeply without an increase
226   in accuracy, Figure 5A. With the exception of these very large rates, these results show that lowering
227   the learning rate does not save energy.
228      Similar to the perceptron, we evaluate how much energy would be required to directly set the
229   synaptic weights to their final values. Traditional learning without synaptic caching is once again
230   energetically inefficient, expending at least $\sim 20$ times more energy compared to this theoretical
231   minimum whatever the desired accuracy level is, Figure 5B. However, by splitting the weights into
232   persistent synaptic weights and transient synaptic caching weights, the network can save substantial
233   amounts of energy. As for the perceptron, depending on the decay and the maintenance cost the
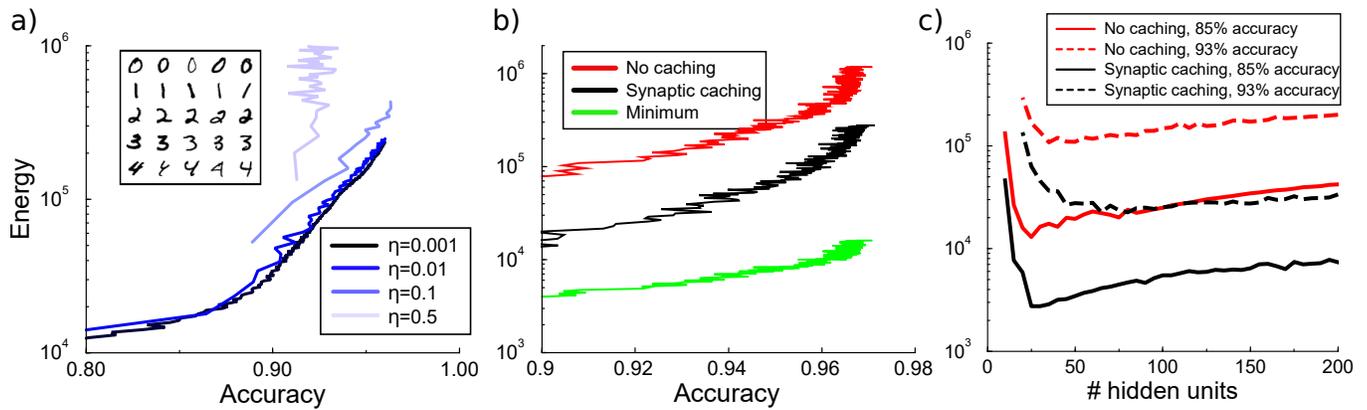
**Figure 5. Energy cost to train a multi-layer back-propagation network to classify digits from the MNIST data set.** (a) Energy rises with the accuracy of identifying the digits from a held-out test data. Except for the larger learning rates, the energy is independent of the learning rate $\eta$. Inset shows some MNIST examples. (b) Comparison of energy required to train the network with/without synaptic caching, and the minimal energy. As for the perceptron and depending on the cost of transient plasticity, synaptic caching can reduce energy need manifold. (c) There is an optimal number of hidden units that minimizes metabolic cost. Both with and without synaptic caching, energy needs are high when the number of hidden units is barely sufficient or very large. Parameters for transient plasticity in (b) and (c): $\eta = 0.1$, $\tau = 1000$, $c = 0.001$.

energy ranges from as little as the minimum to as much as the energy required without caching. Thus the efficiency gain of synaptic caching found for the perceptron carries over to multi-layer networks.

It might seem that smaller networks would be metabolically less costly, because small networks simply contain fewer synapses to modify. On the other hand, we saw above that for the perceptron metabolic costs rise rapidly when cramming many patterns into it. We wondered therefore how energy cost depends on network size in the multi-layer network. Since the number of input units is fixed to the image size and the number of output units equals the ten output categories, we adjust the number of hidden units.

The network fails to reach the desired accuracy if the number of hidden units is too small, Figure 5C. When the network size is barely above the minimum requirement, the network has to compensate the lack of hidden units with longer training time and hence a larger energy expenditure. However, very large networks also require more energy. These results show that from an energy perspective there exists an optimal number of neurons to participate in memory formation. The optimal number depends on the accuracy requirement; as expected, higher accuracies require more hidden units and energy.

## Discussion

Experiments on formation of a long-term memory of a single association suggest that synaptic plasticity is an energetically expensive process. We have shown that energy requirements rise steeply as memory load or designated accuracy level increase. This indicates trade-offs between energy consumption, and network capacity and performance. To improve efficiency we have proposed an algorithm named synaptic caching that temporarily stores changes in the synaptic strength in transient forms of plasticity, and only occasionally consolidates into the persistent forms. Depending on the characteristics (decay and maintenance cost) of transient plasticity, this can lead to large energy savings in the energy required for synaptic plasticity. We stress that from an algorithmic point of view, synaptic caching can be applied to any synaptic learning algorithm (unsupervised, reinforcement, supervised) and does not have specific requirements. Further savings might be possible by adjusting the consolidation threshold as learning progresses and by being pathway-specific (*Leibold and Monsalve-Mercado, 2016*).

The implementation of a consolidation threshold is similar to what has been observed in physiology, in particular in the synaptic tagging and capture literature (*Redondo and Morris, 2011*).

Our results thus give a novel interpretation of those findings. Synaptic consolidation is known to be affected by reward, novelty and punishment (*Redondo and Morris, 2011*), which is compatible with a metabolic perspective as energy is expended only when the stimulus is worth remembering. In addition, our results for instance explain why consolidation is competitive, but transient plasticity is less so (*Sajikumara et al., 2014*), namely the formation of long-term memory is precious. Consistent with this, there is evidence that encouraging consolidation increases energy consumption (*Plaçais et al., 2017*). We also predict that the transient weight changes act as an accumulative threshold for consolidation. That is, sufficient transient plasticity should trigger consolidation, even in the absence of other consolidation triggers. Future characterization of the energy budget of synaptic plasticity should allow more precise predictions of our theory.

Combining persistent and transient storage mechanisms is a strategy well known in traditional computer systems to provide a faster and often energetically cheaper access to memory. In computer systems permanent storage of memories typically requires transmission of all information across multiple transient cache systems until reaching a long-term storage device. The transfer of information is often a bottleneck in computer architectures and consumes considerable power in modern computers (*Kestor et al., 2013*). However, in the nervous system transient and persistent synapses appear to exist next to each other. Local consolidation in a synapse does not require moving information. Using this setup, biology appears to have found a more efficient way to store information.

Memory stability has long fascinated researchers (*Richards and Frankland, 2017*), and in some cases forgetting can be beneficial (*Brea et al., 2014*). Splitting plasticity into transient and persistent forms might prevent catastrophic forgetting in networks (*Leimer et al., 2019*). Here we argue that the main benefit of more transient forms of plasticity is to permit the network to explore the weight space to find a desirable weight configuration using less energy. While this work focuses solely on the metabolic cost of synaptic plasticity, the brain also expends significant amounts of energy on spiking, synaptic transmission, and maintaining resting potential. Learning rules can be designed to reduce costs associated to computation once learning has finished (*Sacramento et al., 2015*). It would be of interest to next understand the precise interaction of computation and plasticity cost during and after learning.

## Methods and Materials

### Energy efficiency of the perceptron

For perceptron we can calculate the energy efficiency of both the classical perceptron and the gain achieved by synaptic caching. We first consider the case that transient plasticity does not decay, as this allows important theoretical simplifications. In the perceptron learning to classify binary patterns Eq. 8, the weight updates are either $+\eta$ or $-\eta$, where $\eta$ is the learning rate, so that the energy spent (Eq. 1, $\alpha = 1$) per update per synapse equals $\eta$. Hence the total energy spent to classify all patterns $M_{\mathrm{perc}} = NK\eta$, where $K$ is the total number of updates. *Opper* (*1988*) showed that learning time diverges as $K \sim (2 - P/N)^{-2}$. We found the numerator numerically to yield $K = 2P/(2 - P/N)^2$.

To calculate the efficiency we compare this to the minimal energy necessary to reach the final weight vector in the perceptron. We approximate the weight trajectory followed by the perceptron algorithm by a random walk. After $K$ updates of step-size $\eta$ the weights approximate a Gaussian distribution with zero mean and variance $K\eta^2$. By short-cutting the random walk, the minimal energy required to reach the weight vector is $M_{\min} = N\langle|w_i|\rangle = \sqrt{\frac{2}{\pi}}\eta N \sqrt{K}$. Hence, we find for the inefficiency (see Figure 1D)

$$\frac{M_{\mathrm{perc}}}{M_{\min}} = \frac{\sqrt{\pi P}}{2 - P/N}.$$

Simulations show that the variance in the weights is actually about 20% smaller than a random walk, likely reflecting correlations in the learning process not captured in the random walk approximation.

312 This explains most of the slight deviation in the ineffeciency between theory and simulation, Fig.1.D.

### Efficiency of synaptic caching

314 To calculate the efficiency gained with synaptic caching we need to calculate both the consolidation
315 energy and the maintenance energy. The consolidation energy equals the number of consolidation
316 events times the size of the updates. The size of the weight updates is equal to the consolidation
317 threshold $\theta$, while the number of consolidation events follows from a random walk argument as
318 $NK/\lceil\theta/\eta\rceil^2$. The ceiling function expresses the fact that when the threshold is smaller than learning
319 rate, consolidation will always occur; we temporarily ignore this scenario. In addition, at the end
320 of learning all remaining transient plasticity is consolidated, which requires an energy $N\langle|s_i(T)|\rangle$.
321 Assuming that the probability distribution of transient weights, $P_s(s)$, has reached steady state at
322 the end of learning, it has a triangular shape (see below) and mean absolute value $\langle|s_i(T)|\rangle = \frac{1}{3}\theta$, so
323 that the total consolidation energy

$$M_{\text{cons}} = \eta^2 \frac{NK}{\theta} + \frac{1}{3}N\theta.$$

324 The energy associated to the transient plasticity is (again assuming that $P_s(s)$ has reached steady
325 state)

$$M_{\text{trans}} = cNT\theta/3, \tag{4}$$

326 where $T$ is the number of time-steps required for learning. We find numerically that $T = \frac{P^{3/2}}{(2-P/N)^2}$.
327 Hence the total energy when using synaptic caching is $M_{\text{cache}} = M_{\text{cons}} + M_{\text{trans}} = N\left[\eta^2 K/\theta + \frac{1}{3}\theta(1+cT)\right]$.
328 The optimal threshold $\hat{\theta}$ is given by $\frac{d}{d\theta}\left[M_{\text{cons}} + M_{\text{trans}}\right] = 0$, or

$$\hat{\theta}^2 = \eta^2 \frac{3K}{1+cT}$$

329 at which the energy is $M_{\text{cache}} = 2\eta N\sqrt{K}\sqrt{1+cT}/\sqrt{3}$. And so the efficiency of synaptic caching is
330 $\frac{M_{\text{cache}}}{M_{\text{min}}} = \sqrt{\frac{2\pi}{3}}\sqrt{1+cT}$. However, as consolidation can maximally occur only once per time-step, $M_{\text{cons}}$
331 cannot exceed $M_{\text{perc}}$ so that the inefficiency is

$$\frac{M_{\text{cache}}}{M_{\text{min}}} = \min\left(\sqrt{\frac{2\pi}{3}(1+cT)}, \sqrt{\frac{\pi}{2}K}\right).$$

332 This equation reasonably matches the simulations, Figure 2C (labeled 'theory').

333 One can include a cost for changing the transient weight, so that $M_{\text{trans}} = c\sum_i \sum_t |s_i(t)| +$
334 $b\sum_i \sum_t |s_i(t+1) - s_i(t)|$, where $b$ codes the cost of making a change. Assuming that consolidat-
335 ing immediately after a weight change does not incur this cost, this yields an extra term in Eq.4 of
336 $bNK(1 - 1/\lceil\theta/\eta\rceil^2)$. Such costs will reduce the efficiency gain achievable by synaptic caching. When
337 $b \geq 1$, it is always cheaper to consolidate.

### Decaying transient plasticity

339 When transient plasticity decays, the situation is more complicated as the learning time depends on
340 the strength of the decay and to our knowledge no analytical expression exists for it. However, it is
341 still possible to estimate the *power*, i.e. the energy per time unit, for both the transient component,
342 denoted $m_{\text{trans}}$, and the consolidation component, $m_{\text{cons}}$. Under the random walk approximation every
343 time the perceptron output does not match the desired output, the transient weight $s_i$ is updated
344 with an amount $\Delta s_i$ drawn from a distribution $Q$, with zero mean and variance $\sigma^2$. Given the update
345 probability $p$, i.e. the fraction of patterns not yet classified correctly, one has $Q_s(\eta) = Q_s(-\eta) = p/2$
346 and $Q_s(0) = 1 - p$, so that $\sigma_s^2 = p\eta^2$. We assume that the synaptic update rate decreases very slowly
347 as learning progresses, hence $p$ is quasi-stationary.

348 Every time-step $\Delta t = 1$ the transient weights decay with a time-constant $\tau$. The synapse is
349 consolidated and $s_i$ is reset to zero whenever the absolute value of the caching weight $|s_i|$ exceeds

350   $\theta$. Given $p$ and $\tau$, we would like to know: 1) how often consolidation events occur which gives
351   consolidation power and 2) the maintenance power $m_{\text{trans}} = cN\langle|s_i|\rangle$. This problem is similar to
352   the random walk to threshold model used for integrate-and-fire neurons, but here there are two
353   thresholds: $\theta$ and $-\theta$.

354   Under the assumptions of small updates and a smooth resulting distribution, the evolution of
355   the probability distribution $P_s(s_i)$ is described by the Fokker-Planck equation, which in the steady
356   state gives

$$0 = -\frac{1}{\tau}\frac{\partial}{\partial s_i}[s_i P_s(s_i)] + \frac{1}{2}\sigma_s^2\frac{\partial^2}{\partial s_i^2}P_s(s_i) + r\delta(s_i).$$

357   The last term is a source term that describes the re-insertion of weights by the reset process. The
358   boundary conditions are $P_s(s_i = \pm\theta) = 0$. While $P_s(s_i)$ is continuous in $s_i$, the source introduces a
359   cusp in $P_s(s_i)$ at the reset value. Conservation of probability ensures that $r$ equals the outgoing flux
360   at the boundaries. One finds

$$P_s(s_i) = \frac{1}{Z}\exp\left[-\frac{s_i^2}{\sigma^2}\right]\left[\operatorname{erfi}\left(\frac{|s_i|}{\sigma}\right) - \operatorname{erfi}\left(\frac{\theta}{\sigma}\right)\right],$$

where $\operatorname{erfi}(x) = -i\operatorname{erf}(ix)$, $\sigma^2 = \frac{\tau}{\Delta t}\sigma_s^2$ and with normalization factor

$$Z = \frac{2\theta^2}{\sqrt{\pi}\sigma}\,_2F_2\left(1,1;\frac{3}{2},2;-(\frac{\theta}{\sigma})^2\right) - \sqrt{\pi}\sigma\operatorname{erf}\left(\frac{\theta}{\sigma}\right)\operatorname{erfi}\left(\frac{\theta}{\sigma}\right),$$

361   where $_2F_2$ is the generalized hypergeometric function. (In the limit of no decay this becomes a
362   triangular distribution $P_s(s_i) = [\theta - |s_i|]/\theta^2$.)

We obtain maintenance power

$$m_{\text{trans}} = cN\langle|s_i|\rangle \tag{5}$$

$$= \frac{cN}{Z}\left[\frac{2\theta\sigma}{\sqrt{\pi}} - \sigma^2\operatorname{erfi}\left(\frac{\theta}{\sigma}\right)\right]. \tag{6}$$

363   For small $\theta/\sigma$, i.e. small decay, this is linear in $\theta$, $m_{\text{trans}} \approx \frac{cN\theta}{3}$. It saturates for large $\theta$ because then
364   the decay dominates and the threshold is hardly ever reached.

The consolidation rate follows from Fick's law

$$r = \frac{1}{2}\sigma^2 P_s'(-\theta) - \frac{1}{2}\sigma^2 P_s'(\theta)$$

$$= \frac{-2\sigma}{Z\sqrt{\pi}}.$$

365   The consolidation power is

$$m_{\text{cons}} = N\theta r. \tag{7}$$

366   In the limit of no decay one has $r = \sigma^2/\theta^2$, so that $m_{\text{cons}} = pN\eta^2/\theta$. Strictly speaking this approxi-
367   mates learning with a random walk process and assumes local consolidation, Figure 4A. However,
368   Eqs. 6 and 7 give a good prediction of the simulation when provided with the time-varying update
369   probability from the simulation, Figure 6.

## Simulations

### Perceptron

372   Unless stated otherwise, we use a perceptron with $N = 1000$ input units to classify $P = N$ random
373   binary ($\pm 1$ with equal probability) input patterns $\mathbf{x}^{(p)}$, each to be associated to a randomly assigned
374   desired binary output $d^{(p)}$. Each input unit is connected with a weight $w_i$ signifying the strength
375   of the connection. An 'always-on' bias unit with corresponding weight is included to adjust the
376   threshold of the perceptron. The perceptron output $y$ of a pattern is determined by the Heaviside
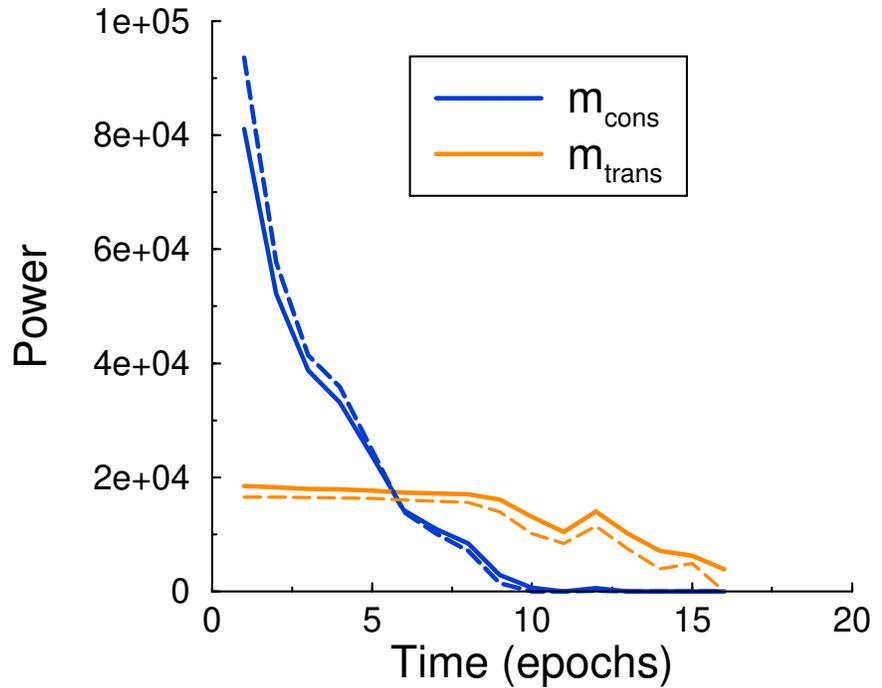
**Figure 6. Maintenance and consolidation power.** Power (energy per epoch) of the perceptron vs epoch. Solid curves are from simulation, dashed curves are the theoretical predictions, Eqs. 6 and 7, with $\sigma$ calculated by using the perceptron update rate $p$ extracted from the simulation. Both powers are well described by the theory. Parameters: $\tau = 500$, $c = 0.01$, $\theta = 5$.

377    step function $\Theta$, $y = \Theta(\mathrm{w}.\boldsymbol{x})$. If for a given pattern $p$, the output does not match the desired pattern
378    output, $\boldsymbol{w}$ is adjusted according to

$$\Delta w_i = \eta \left( d^{(p)} - y^{(p)} \right) x_i^{(p)}, \tag{8}$$

379    where the learning rate $\eta$ can be set to one without loss of generality. The perceptron algorithm
380    cycles through all patterns until classified correctly. In principle the magnitude of the weight vector,
381    and hence the minimal energy, can be arbitrarily small for a noise-free binary perceptron. However,
382    this paradox is resolved as soon as robustness to any post-synaptic noise is required.

383    Multi-layer networks
384    For the multi-layer networks trained on MNIST, we use networks with one hidden layer, logistic
385    units, and one-hot encoding at the output. Weights are updated according to the mean squared
386    error back-propagation rule without regularization.
387      Simulation scripts for both the perceptron and the multilayer network can be found at https:
388    //github.com/vanrossumlab/li_vanrossum_19.

393    **References**
394    Alle, H., Roth, A., and Geiger, J. R. P. (2009). Energy-efficient action potentials in hippocampal mossy fibers.
395      *Science*, 325(5946):1405–1408.

Attwell, D. and Laughlin, S. B. (2001). An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145.

Azari, N. P. (1991). Effects of glucose on memory processes in young adults. *Psychopharmacology*, 105(4):521–524.

Barrett, A. B., Billings, G. O., Morris, R. G. M., and van Rossum, M. C. W. (2009). State based model of long-term potentiation and synaptic tagging and capture. *PLoS Computational Biology*, 5(1):e1000259.

Brea, J., Urbanczik, R., and Senn, W. (2014). A normative theory of forgetting: lessons from the fruit fly. *PLoS Computational Biology*, 10(6):e1003640.

Brunel, N. (2016). Is cortical connectivity optimized for storing information? *Nature Neuroscience*, 19(5).

Brunel, N., Hakim, V., Isope, P., Nadal, J.-P., and Barbour, B. (2004). Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell. *Neuron*, 43(5):745–757.

Costa, R. P., Froemke, R. C., Sjöström, P. J., and van Rossum, M. C. W. (2015). Unified pre- and postsynaptic long-term plasticity enables reliable and flexible learning. *eLife*, 4:e09457.

Craft, S., Murphy, C., and Wemstrom, J. (1994). Glucose effects on complex memory and nonmemory tasks: the influence of age, sex, and glucoregulatory response. *Psychobiology*, 22(2):95–105.

D'Souza, P., Liu, S.-C., and Hahnloser, R. H. R. (2010). Perceptron learning rule derived from spike-frequency adaptation and spike-time-dependent plasticity. *PNAS*, 107(10):4722–4727.

Fohlmeister, J. F. (2009). A nerve model of greatly increased energy-efficiency and encoding flexibility over the hodgkin–huxley model. *Brain Research*, 1296:225–233.

Frey, U. and Morris, R. G. (1997). Synaptic tagging and long-term potentiation. *Nature*, 385(6616):533–536.

Gardner, E. J. (1987). Maxium storage capaity in neural network models. *Europhysics Letters*, 4:481–485.

Gold, P. E. (1986). Glucose modulation of memory storage processing. *Behavioral and Neural Biology*, 45(3):342–349.

Hafner, A.-S., Donlin-Asp, P. G., Leitch, B., Herzog, E., and Schuman, E. M. (2019). Local protein synthesis is a ubiquitous feature of neuronal pre- and postsynaptic compartments. *Science*, 364(6441):eaau3644.

Hall, J. L., Gonder-Frederick, L., Chewning, W., Silveira, J., and Gold, P. (1989). Glucose enhancement of performance of memory tests in young and aged humans. *Neuropsychologia*, 27(9):1129–1138.

Harris, J. J., Jolivet, R., and Attwell, D. (2012). Synaptic energy use and supply. *Neuron*, 75(5):762–777.

Jaumann, S., Scudelari, R., and Naug, D. (2013). Energetic cost of learning and memory can cause cognitive impairment in honeybees. *Biology Letters*, 9(4):20130149.

Kestor, G., Gioiosa, R., Kerbyson, D. J., and Hoisie, A. (2013). Quantifying the energy cost of data movement in scientific applications. *IEEE International Symposium on Workload Characterization (IISWC)*, pages 56–65.

Leibold, C. and Monsalve-Mercado, M. M. (2016). Asymmetry of neuronal combinatorial codes arises from minimizing synaptic weight change. *Neural Computation*, 28(8):1527–52.

Leimer, P., Herzog, M., and Senn, W. (2019). Synaptic weight decay with selective consolidation enables fast learning without catastrophic forgetting. *bioRxiv*.

Lennie, P. (2003). The cost of cortical computation. *Current Biology*, 13(6):493–497.

Levy, W. B. and Baxter, R. A. (1996). Energy efficient neural codes. *Neural Computation*, 8(3):531–543.

Levy, W. B. and Baxter, R. A. (2002). Energy-efficient neuronal computation via quantal synaptic failures. *Journal of Neuroscience*, 22(11):4746–4755.

Mery, F. and Kawecki, T. J. (2005). A cost of long-term memory in drosophila. *Science*, 308(5725):1148.

Opper, M. (1988). Learning times of neural networks: Exact solution for a perceptron algorithm. *Physical Review A*, 38(7):3824–3826.

Padamsey, Z. and Emptage, N. (2014). Two sides to long-term potentiation: a view towards reconciliation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1633):20130154.

Plaçais, P.-Y., de Tredern, È., Scheunemann, L., Trannoy, S., Goguel, V., Han, K.-A., Isabel, G., and Preat, T. (2017). Upregulated energy metabolism in the drosophila mushroom body is the trigger for long-term memory. *Nature Communications*, 8(15510).

Plaçais, P.-Y. and Preat, T. (2013). To favor survival under food shortage, the brain disables costly memory. *Science*, 339(6118):440–442.

Potter, W. B., O'Riordan, K. J., Barnett, D., Osting, S. M., Wagoner, M., Burger, C., and Roopra, A. (2010). Metabolic regulation of neuronal plasticity by the energy sensor AMPK. *PloS one*, 5(2):e8996.

Redondo, R. L. and Morris, R. G. M. (2011). Making memories last: the synaptic tagging and capture hypothesis. *Nature Reviews Neuroscience*, 12(1):17–30.

Richards, B. A. and Frankland, P. W. (2017). The persistence and transience of memory. *Neuron*, 94(6):1071–1084.

Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Spartan Books.

Sacramento, J., Wichert, A., and van Rossum, M. C. W. (2015). Energy efficient sparse connectivity from imbalanced synaptic plasticity rules. *PLoS Computational Biology*, 11(6):e1004265.

Sajikumar, S., Navakkode, S., Sacktor, T. C., and Frey, J. U. (2005). Synaptic tagging and cross-tagging: the role of protein kinase Mzeta in maintaining long-term potentiation but not long-term depression. *Journal of Neuroscience*, 25(24):5750–5756.

Sajikumara, S., Morris, R. G. M., and Korte, M. (2014). Competition between recently potentiated synaptic inputs reveals a winner-take-all phase of synaptic tagging and capture. *PNAS*, 111(33):12217–12221.

458
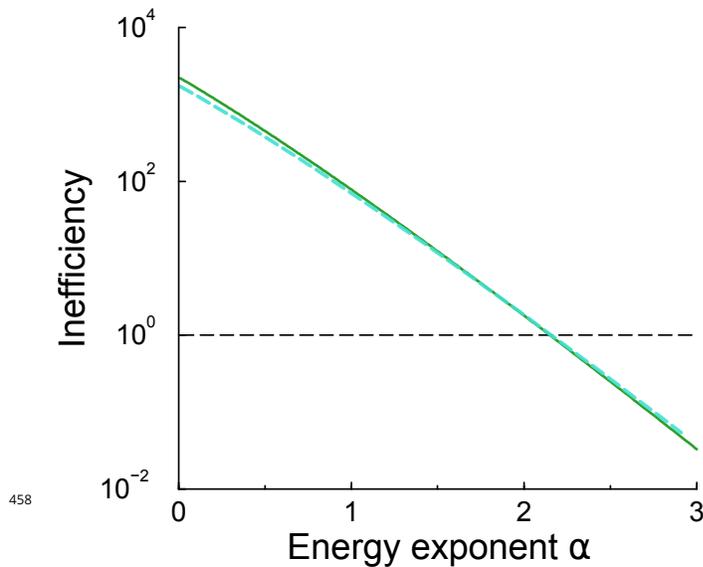
**Figure 1–Figure supplement 1. The energy inefficiency of perceptron learning for various energy variants.** The energy inefficiency of perceptron learning when the energy associated to synaptic update is $\Sigma_{i,t}|w_i(t) - w_i(t-1)|^\alpha$ and the exponent $\alpha$ is varied (green curve). The case $\alpha = 1$ is used throughout the main text. The inefficiency is the ratio between the energy needed to train the perceptron and the energy required to set the weights directly to their final value. When $\alpha = 0$, the energy is equal to the number of updates made. When $\alpha = 1$, the energy is the sum of individual update amounts. When $\alpha > 1$ it costs less energy to make many small weight updates compared to one large one. When $\alpha \gtrsim 2$, this effect is so strong that even the random walk of the perceptron is less costly than directly setting the weights to their final value. We consider $0 \leq \alpha \leq 1$ to be the biologically relevant regime. Also shown is the inefficiency when only potentiation costs energy, and depression comes at no cost i.e. $M = \Sigma_{i,t}\left[w_i(t) - w_i(t-1)\right]^\alpha_+$ (overlapping cyan curve). This has virtually identical (in)efficiency.
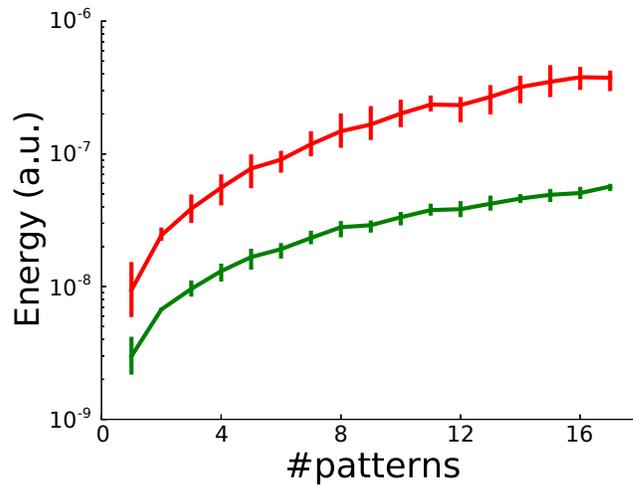
**Figure 2–Figure supplement 1. Synaptic caching in a spiking neuron with a biologically plausible perceptron-like learning rule.** To demonstrate the generality of our results, independent of learning rule or implementation, we implement a spiking biophysical perceptron. *D'Souza et al.* (*2010*) proposed perceptron-like learning by combining synaptic spike-time dependent plasticity (STDP) with spike-frequency adaptation (SFA). In their model the leaky integrate-and-fire neuron receives auditory input and delayed visual input. The neuron's objective is to balance its auditory response $A = w \cdot x$ to its visual response $V$ by adjusting the weights $w$ of its auditory synapses through STDP. The visual input is the supervisory signal. We use 100 auditory inputs, and measure the energy for the neuron to learn $w$ so that each auditory input pattern becomes associated to a (binary) visual input. We repeatedly present patterns $x^{(p)}$, each with two activated auditory inputs until $w$ stabilized as D'Souza et al. The training is considered successful if the auditory responses of all the input patterns associated to the same binary visual input fall within two standard deviations from the mean auditory response of those patterns, and are at least five standard deviations away from the mean auditory response of other patterns. Synaptic caching is implemented as in the main text by splitting $w$ into persistent forms and transient forms. We consider the optimal scenario where the transient weights do not decay and have no maintenance cost. Also in the biophysical implementation of perceptron learning, synaptic caching (green curve) saves a significant amount of energy compared to without caching (red curve), suggesting that synaptic caching works universally regardless of learning algorithm or biophysical implementation.
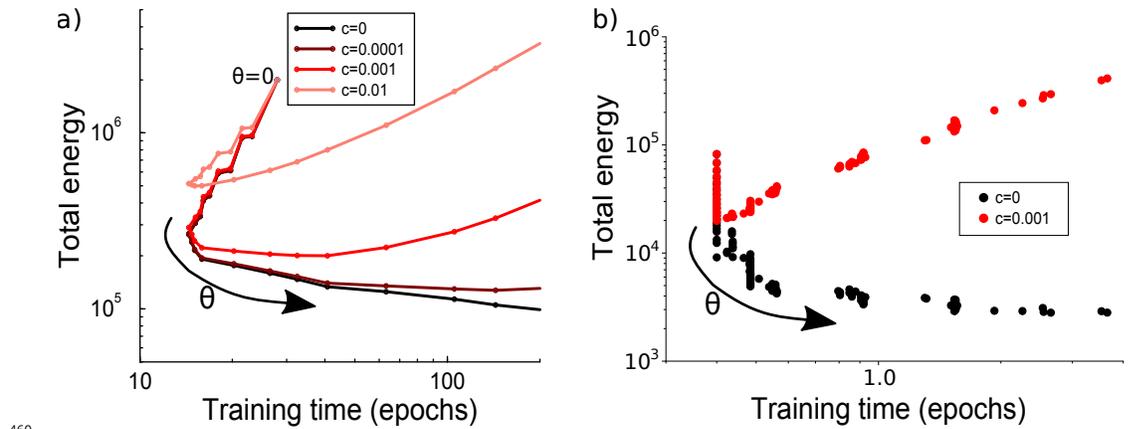
**Figure 3–Figure supplement 1. The effects of consolidation threshold on energy cost and learning time.** (a) Parametric plot of learning time vs energy while the consolidation threshold $\theta$ is varied. The threshold value runs from 0 to 10 in steps of 0.5. For small maintenance costs, the threshold determines a trade-off between either a short learning time or a low energy (e.g. black curve). At higher maintenance costs, the most energy efficient threshold also leads to a short learning time. Average over 100 runs; parameter: $\tau = 10^3$. (b) Similar to the perceptron results in panel a, the effects of consolidation threshold on energy cost and learning time for training in a multi-layer network vary depending on the maintenance cost $c$. Here, the threshold starts at 0.005 and is in increments of 0.005. When $c = 0$ (black dots, each representing a unique consolidation threshold), there is a trade-off between shorter learning time and lower energy cost. When $c = 0.001$ (red dots), the result is similar to the perceptron result with $c = 0.01$, where optimizing learning time or energy cost leads to a similar threshold. Parameters: $\eta = 0.1$, $\tau = 10^4$, required accuracy $= 0.93$.