

# A contrastive rule for meta-learning

Nicolas Zucchet<sup>\*,†</sup>, Simon Schug<sup>\*</sup>, Johannes von Oswald<sup>\*</sup>,  
Dominic Zhao and João Sacramento<sup>†</sup>

*Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland*

## Abstract

Meta-learning algorithms leverage regularities that are present on a set of tasks to speed up and improve the performance of a subsidiary learning process. Recent work on deep neural networks has shown that prior gradient-based learning of meta-parameters can greatly improve the efficiency of subsequent learning. Here, we present a biologically plausible meta-learning algorithm based on equilibrium propagation. Instead of explicitly differentiating the learning process, our contrastive meta-learning rule estimates meta-parameter gradients by executing the subsidiary process more than once. This avoids reversing the learning dynamics in time and computing second-order derivatives. In spite of this, and unlike previous first-order methods, our rule recovers an arbitrarily accurate meta-parameter update given enough compute. We establish theoretical bounds on its performance and present experiments on a set of standard benchmarks and neural network architectures.

## 1 Introduction

When posed with a sequence of tasks that bear some relation to one another, a learning system that is capable of meta-learning will be able to improve its performance as the number of tasks increases. A successful meta-learner discovers shared structure across tasks and modifies its own inductive bias accordingly [1, 2]. Eventually, the system is able to generalize to new problem instances given little additional data. Such ability is believed to be a hallmark of human intelligence [3], and it has been observed in non-human primates as well as in other animal species [4]. Determining which mechanisms support this form of learning in the brain is a fundamental question in neuroscience [5].

A well-known approach to endow a machine learning system with the capability to meta-learn is to introduce two distinct learning timescales. Some of the parameters, designated as meta-parameters, are updated across tasks but kept fixed while solving any single problem. Learning both the fast-evolving, lower-level parameters and the slowly-evolving meta-parameters from data can be formulated as a hierarchical Bayesian inference problem [1, 6–9] or as an empirical risk minimization problem [10–12]. This approach has led to recent great successes when applied to deep neural network models [13]. Notably, meta-learning has enabled such large nonlinear models to learn and generalize well from datasets comprising just a handful of examples [14].

In deep learning, meta-parameters are generally optimized by stochastic gradient descent. This requires differentiating the lower-level learning process, which is often once again a variant of gradient descent [10]. Differentiating a long chain of lower-level parameter updates by backpropagation is typically too costly as high-dimensional intermediate parameter states have to be stored. In practice, this difficulty is circumvented either by restricting the lower-level learning process to a small number of updates or by truncating the backpropagation through training procedure [15]. Entirely truncating backpropagation yields a first-order method [10].

We propose an alternative algorithm to learn meta-parameters which does not explicitly differentiate the lower-level learning rule. Instead, we resort to the equilibrium propagation theorem found and

<sup>\*</sup>N.Z., S.S. and J.v.O. contributed equally to this work.

<sup>†</sup>To whom correspondence may be addressed: [nzucchet@student.ethz.ch](mailto:nzucchet@student.ethz.ch), [sacramento@ini.ethz.ch](mailto:sacramento@ini.ethz.ch)

proved by Scellier and Bengio [16]. When applied to learn the weights of a Hopfield network, equilibrium propagation gives rise to a variant of the well-known contrastive Hebbian learning rule [17–21]. Here, we exploit an equivalence between the training of energy-based neural networks and implicit meta-learning and show that, when applied to meta-learning problems, equilibrium propagation gives rise to a new meta-parameter learning rule. We term this rule contrastive meta-learning.

Unlike conventional backpropagation through training, our contrastive meta-learning rule does not require storing and revisiting a parameter trajectory in reverse time order, it does not require computing Hessian-vector products, and it does not depend on the optimizer used to update lower-level parameters. Unlike implicit differentiation methods [22–27], contrastive meta-learning only uses first-order derivatives of an objective function. The algorithm allows controlling the error incurred when estimating the exact meta-gradient at the expense of additional compute, unlike previous first-order algorithms [10, 25, 28]. These properties make our contrastive meta-learning a candidate rule for biologically-plausible meta-learning. Nonetheless, our algorithm still requires differentiating an objective function, while being agnostic as to how such derivatives are actually estimated. How the brain could efficiently approximate derivatives of an objective function with respect to a set of high-dimensional parameters has been the subject of intense recent research [16, 29–40].

We first derive the contrastive meta-learning rule and present a mathematical analysis of the meta-gradient estimation error which describes the behavior of any learning rule based on equilibrium propagation. This analysis is followed by an experimental evaluation of the contrastive meta-learning rule on increasingly difficult benchmarks, from low-dimensional problems where we can confirm our theory by comparing to exact meta-gradients, to the Omniglot [14] few-shot learning challenge, that we approach with meta-learned deep neural network models. We conclude with a discussion of the implications of our findings to neuroscience.

## 2 Contrastive meta-learning

**Problem setting.** We consider bilevel optimization problems of the following form:

$$\begin{aligned} \min_{\theta} L^{\text{out}}(\phi_{\theta}^*, \theta) \\ \text{s.t. } \phi_{\theta}^* \in \arg \min_{\phi} L^{\text{in}}(\phi, \theta), \end{aligned} \tag{1}$$

where  $\theta$  denotes the vector of meta-parameters,  $\phi$  the vector of lower-level parameters,  $L^{\text{in}}(\phi, \theta)$  the inner loss function and  $L^{\text{out}}(\phi, \theta)$  the outer loss function. We therefore seek a meta-parameter configuration  $\theta^*$  such that, after minimizing the inner loss function  $L^{\text{in}}$  while holding meta-parameters constant, the outer loss  $L^{\text{out}}$  is at its minimum. We consider the case where  $L^{\text{in}}$  explicitly depends on the two kinds of parameters  $\phi$  and  $\theta$ , while  $L^{\text{out}}$  might depend only on  $\phi$ . The subscript notation  $\phi_{\theta}^*$  emphasizes the dependence of the inner problem solution on the meta-parameters  $\theta$ .

Eq. 1 captures a wide range of meta-learning problems. For instance,  $L^{\text{in}}$  and  $L^{\text{out}}$  could comprise the same loss functional evaluated on different data, possibly with additional regularization terms. This yields the cross-validation criterion commonly employed for supervised meta-learning [13] and is the setting that we consider in our experiments. As an additional example,  $L^{\text{in}}$  could be set to an unsupervised learning criterion and  $L^{\text{out}}$  to a supervised one, following another line of recent work [41].

Meta-learning problems that can be expressed as Eq. 1 have been extensively studied and solved using implicit differentiation techniques when  $\theta$  and  $\phi$  parameterize neural network models [22–27]. The assumption that the inner problem is solved to equilibrium can be contrasted to the case where  $\phi$  is updated with a single step of gradient descent [10]. The two cases sit at two opposing ends of a spectrum, but both simplify the process of determining the meta-gradient. The cost of backpropagating through training scales with the number of updates and is therefore minimal for one step. Conversely, the assumption that the parameters are at a fixed point  $\phi_{\theta}^*$  allows invoking the equilibrium propagation theorem avoiding backpropagation through training altogether, as we show next.

**Meta-learning by equilibrium propagation.** Equilibrium propagation was originally introduced in the context of supervised learning of energy-based recurrent neural networks, whose neural dynamics minimize an energy function. This problem can be recovered by setting  $L^{\text{in}}$  to the energy function and

$L^{\text{out}}$  to a cost function imposed on output neurons; the neural activity plays the role of our lower-level parameters  $\phi$ , and recurrent network weights are our meta-parameters  $\theta$ . Meta-learning problems of the form of Eq. 1 can therefore be approached with equilibrium propagation. We note that the link between general recurrent neural network learning and meta-parameter optimization precedes our paper and can be traced back at least to ref. [7].

We review the main equilibrium propagation result below. We define the augmented loss

$$\mathcal{L}(\phi, \theta, \beta) = L^{\text{in}}(\phi, \theta) + \beta L^{\text{out}}(\phi, \theta), \quad (2)$$

where  $\beta \in \mathbb{R}$  is an auxiliary scalar parameter called the nudging strength. A finite  $\beta$  mixes the two loss functions; when  $\beta$  is zero only the inner loss survives. Denoting by  $\phi_{\theta, \beta}^*$  a minimizer of  $\mathcal{L}(\cdot, \theta, \beta)$ , our goal is to estimate the meta-gradient  $\nabla_{\theta} := d_{\theta} L^{\text{out}}(\phi_{\theta, 0}^*, \theta)$ . Throughout this manuscript, we use the symbol  $d$  for total derivatives and  $\partial$  for partial derivatives. We are now in position to restate the equilibrium propagation theorem.

**Theorem 1** (Scellier and Bengio [16]). *If  $\phi_{\theta, \beta}^*$  is a minimizer of  $\mathcal{L}(\cdot, \theta, \beta)$ , then*

$$\frac{d}{d\beta} \frac{\partial \mathcal{L}}{\partial \beta}(\phi_{\theta, \beta}^*, \theta, \beta) = \frac{d}{d\beta} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_{\theta, \beta}^*, \theta, \beta). \quad (3)$$

For Theorem 1 to hold certain conditions must be met, that we review in Appendix A.

Since  $\partial_{\beta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta) = L^{\text{out}}(\phi_{\theta, \beta}^*, \theta)$ , when  $\beta$  is zero the left-hand side of Eq. 3 corresponds to the meta-gradient  $\nabla_{\theta}$  that we seek. Thus, thanks to Theorem 1, the problem of determining  $\nabla_{\theta}$  has been reduced to that of evaluating the right-hand side of Eq. 3, a derivative of a vector-valued function with respect to a scalar variable. Scellier and Bengio [16] propose to estimate this quantity using a finite difference method. This yields the meta-gradient estimate

$$\widehat{\nabla}_{\theta} = \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_{\beta}, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_0, \theta, 0) \right). \quad (4)$$

In the equation above, the first-phase approximate solution  $\hat{\phi}_0$  is an approximation to a minimum  $\phi_0^*$  of the inner loss function  $L^{\text{in}}$ . The second-phase approximate solution  $\hat{\phi}_{\beta}$  is an approximation to a minimum  $\phi_{\beta}^*$  of the augmented loss  $\mathcal{L}$  when the nudging strength takes a non-zero value  $\beta$ , obtained consecutively by initializing the minimization procedure at  $\hat{\phi}_0$ . The meta-gradient  $\nabla_{\theta}$  is recovered when the inner optimization problems are exactly solved, i.e.  $\hat{\phi}_0 = \phi_0^*$  and  $\hat{\phi}_{\beta} = \phi_{\beta}^*$ , and  $\beta$  goes to zero. We discuss and analyze meta-gradient estimators based on more refined finite difference methods in Appendix D.

**The contrastive meta-learning rule.** Eq. 4 provides a learning rule to update meta-parameters. A remarkable property of this rule is that it only uses gradients of the loss function  $\mathcal{L}$ ; no second-order derivatives are involved. Such gradients can be efficiently computed by backpropagation of error or biologically-plausible approximations thereof. In principle, even equilibrium propagation can be invoked in a nested fashion to approximate the required gradients. Furthermore, the learning rule is causal. Instead of backpropagating through a learning process, Eq. 4 contrasts meta-parameter gradients, measured at two moments in time. This process can be thought of as the meta-learning analogue of contrastive Hebbian learning, which compares neural activity instead. When applied to meta-learning problems we henceforth refer to Eq. 4 as the contrastive meta-learning rule.

### 3 Theoretical analysis

The contrastive meta-learning rule (Eq. 4) only provides an approximation to the meta-gradient. This approximation can be improved with additional computation, by decreasing the nudging strength  $\beta$  and refining the lower-level learning process. We now analyze the impact of such a refinement on the quality of the meta-gradient estimate.

The first step in this analysis involves upper bounding the meta-gradient estimation error, given the value of  $\beta$  and the error made in the approximation of the fixed points of the lower-level learning

process. Two conflicting phenomena impact the estimation error. First, our meta-learning rule is based on potentially inexact fixed points. Second, the finite difference approximation of the  $\beta$ -derivative yields the so-called finite difference error. Informally, higher  $\beta$  values will reduce the sensitivity to crude approximations to the lower-level solutions while increasing the finite difference error. Theorem 2 theoretically justifies this intuition under the idealized regime of strong convexity and smoothness defined in Assumption 1. This result holds for every rule induced by equilibrium propagation.

**Assumption 1.** Assume that  $L^{\text{in}}$  and  $L^{\text{out}}$  are three-times continuously differentiable and that, as functions of  $\phi$ :

- i.  $\partial_{\theta} L^{\text{in}}$  and  $\partial_{\theta} L^{\text{out}}$  are Lipschitz continuous.
- ii.  $L^{\text{in}}$  and  $L^{\text{out}}$  are strongly convex, smooth<sup>1</sup>.
- iii. the Hessians of  $L^{\text{in}}$  and  $L^{\text{out}}$  are Lipschitz-continuous.
- iv.  $\partial_{\phi} \partial_{\theta} L^{\text{in}}$  and  $\partial_{\phi} \partial_{\theta} L^{\text{out}}$  are Lipschitz continuous.

**Theorem 2.** Let  $\beta > 0$  and  $(\delta, \delta')$  such that  $\|\phi_{\theta,0}^* - \hat{\phi}_0\| \leq \delta$  and  $\|\phi_{\theta,\beta}^* - \hat{\phi}_{\beta}\| \leq \delta'$ . Under Assumption 1, there exists a  $\theta$ -dependent constant  $C$  such that

$$\|\nabla_{\theta} - \hat{\nabla}_{\theta}\| \leq C \left( \frac{\delta + \delta'}{\beta} + \delta' + \frac{\beta}{1 + \beta} \right) =: \mathcal{B}(\delta, \delta', \beta).$$

If we additionally assume that  $\theta$  lies in a compact set, we can choose  $C$  to be independent of  $\theta$ .

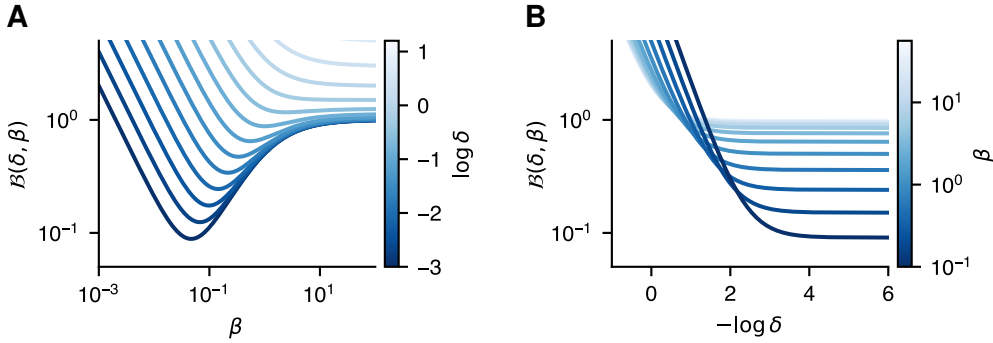


Figure 1: Visualization of the bound  $\mathcal{B}$  from Theorem 2 ( $C = 1$ ,  $\delta = \delta'$ ), as a function of  $\beta$  (A) and as a function of  $\delta = \delta'$  (B). Decreasing  $\beta$  improves the quality of the estimation, but only if it is associated to better approximations in the two phases.

We visualize our bound in Fig. 1, as a function of  $\beta$  and of the fixed point approximation errors  $\delta$  and  $\delta'$ . When  $\delta$  and  $\delta'$  are fixed, the estimation error quickly increases when  $\beta$  deviates from its optimal value and it caps for large  $\beta$  values. A better fixed point approximation naturally improves the quality of the meta-gradient estimate for  $\beta$  held constant. However, the benefits saturate above some  $\beta$ -dependent value: investing extra compute in the approximation of the fixed point does not pay off if  $\beta$  is not decreased accordingly.

Assumption 1 is not as restrictive as it may seem. For the function  $\phi_{\theta,\beta}^*$  to be defined in a neighborhood of  $(\theta, \beta = 0)$ , and therefore for the equilibrium propagation theorem to hold,  $L^{\text{in}}(\cdot, \theta)$  needs to be steep around  $\phi_{\theta,0}^*$ . This condition turns out to be equivalent to a local strong convexity property. The other assumptions also hold in the vicinity of  $\phi_{\theta,0}^*$ . Our assumptions therefore extend local properties that follow from equilibrium propagation assumptions to the entire lower-level parameter space.

Theorem 2 highlights the importance of considering  $\beta$  as a hyperparameter of the learning rule that needs to be adjusted to yield the best possible meta-gradient estimate. Corollary 1 removes the dependence in  $\beta$  and considers the best achievable bound under given fixed point approximation errors.

<sup>1</sup>A function  $f$  is smooth if its gradient is Lipschitz continuous. In this case, it is equivalent to the Lipschitz continuity of  $\partial_{\phi} L^{\text{in}}$  and  $\partial_{\phi} L^{\text{out}}$ .

**Corollary 1.** *Under Assumption 1, if we suppose that for every strictly positive  $\beta$  we approximate the two fixed points with precision  $\delta$  and  $\delta'$  and if  $(\delta + \delta') < C/B^{\text{in}}$ , the best achievable bound in Theorem 2 is smaller than*

$$B^{\text{out}}\delta' + 2\sqrt{CB^{\text{in}}(\delta + \delta')}$$

and is attained for  $\beta$  equal to

$$\beta^*(\delta, \delta') = \frac{\sqrt{B^{\text{in}}(\delta + \delta')}}{\sqrt{C} - \sqrt{B^{\text{in}}(\delta + \delta')}}.$$

The limiting part of the bound depends on  $\delta + \delta'$  and not only on one of the two quantities, suggesting that the two errors should be of the same magnitude to avoid unnecessary computations.

We prove Theorem 2 and Corollary 1 in Appendix C.

## 4 Experiments

We next test our contrastive meta-learning rule on standard supervised meta-learning problems. The objective is twofold: confirm our theoretical results and compare the performance of our rule to other first-order methods (when applicable) as well as to more sophisticated techniques which rely on implicit differentiation and backpropagation through training.

### 4.1 Experimental setup

In our experiments, we consider inner and outer loss functions of the following form:

$$\begin{aligned} L^{\text{in}}(\phi, \theta) &= L(\phi, \theta, \mathcal{D}^{\text{train}}) + R(\phi, \theta), \\ L^{\text{out}}(\phi, \theta) &= L(\phi, \theta, \mathcal{D}^{\text{val}}), \end{aligned} \tag{5}$$

where  $L$  measures the expected discrepancy between the predictions of a neural network and the target outputs on a dataset  $\mathcal{D}$ ,  $\mathcal{D}^{\text{train}}$  is some training set,  $\mathcal{D}^{\text{val}}$  is a heldout validation set and  $R(\phi, \theta)$  is a regularizer. Once plugged into the bilevel optimization problem of Eq. 1, we have an instance of cross-validation-based meta-learning, with the goal of improving generalization [7, 22–24, 27].

The cross-validation-based learning setting presented above only considers one task. The formulation can be extended to several tasks and few-shot learning, where each task  $\tau$  consists of small training  $\mathcal{D}_{\tau}^{\text{train}}$  and validation  $\mathcal{D}_{\tau}^{\text{val}}$  sets. The inner  $L^{\text{in}}$  and outer  $L^{\text{out}}$  losses defined in Eq. 5 then become task-dependent through the data on which they are evaluated; we now denote them  $L_{\tau}^{\text{in}}$  and  $L_{\tau}^{\text{out}}$ . Implicit few-shot learning can then be formalized as:

$$\begin{aligned} \min_{\theta} \mathbb{E}_{\tau} [L_{\tau}^{\text{out}}(\phi_{\theta, \tau}^*, \theta)] \\ \text{s.t. } \phi_{\theta, \tau}^* \in \arg \min_{\phi} L_{\tau}^{\text{in}}(\phi, \theta). \end{aligned} \tag{6}$$

In practice, we solve this optimization problem by sampling tasks and computing the corresponding meta-gradients, thus going back to Eq. 1 for which we have introduced our meta-learning rule.

### 4.2 Models

To complete the presentation of our experimental setting, we finally specify how  $\phi$  and  $\theta$  interact within our neural networks and regularizers. We introduce two different models below.

The first meta-parameterization we consider is a simple abstraction of the internal state of a biological synapse, which can exhibit synaptic adaptation and consolidation effects that cannot be captured by a single weight variable [42]. Following [43, 44] we model these effects through a quadratic regularizer with synapse-specific parameters. This regularizer can be given a Bayesian interpretation as a Gaussian weight prior [7].

In this complex synapse model, besides its actual weight  $\phi_i$ , a synapse is endowed with two meta-parameters, a slow weight  $\omega_i$  towards which the fast weight  $\phi_i$  is attracted and a meta-plasticity parameter  $\lambda_i$  which determines the strength of this attractive force. The interaction between these three components is modeled by a quadratic penalty:

$$R(\phi, \theta) = \frac{1}{2} \sum_{i=1}^N \lambda_i (\phi_i - \omega_i)^2, \quad (7)$$

where  $N$  is the number of synapses. Furthermore, the data-dependent loss  $L$  (cf. Eq. 5) depends only on  $\theta$ , and not on  $\phi$ . Applied to this model, the goal of our contrastive meta-learning rule is to improve generalization and speed-up synaptic learning by appropriately changing the internal synaptic state variables  $\omega$  and  $\lambda$ . Changing the internal state variables on the slow (task-to-task) timescale of meta-learning can be thought of as changing the rules of the actual plasticity process which adapts synaptic weights. Contrastive meta-learning yields a meta-learning rule which only uses information that is local to the synapse. This rule is described in Appendix B.

As an alternative, we consider a second meta-parameterization where the gain and threshold of the neural input-output transfer function are the only adjustable lower-level parameters, together with the weights of the last readout layer which solves the task at hand. We call this the learning-by-modulation model. In practice, we take the (multiplicative) gain and (additive) threshold parameters of batch normalization units as our lower-level parameters. In this model, the data-dependent loss term  $L$  in Eq. 5 depends on both  $\theta$  and  $\phi$ . A similar model has been studied by Zintgraf et al. [45]. With this parameterization, the vast majority of synaptic weights play the role of meta-parameters; the dimension of the parameters that are learned is of the order of the number of neurons, not synapses. While this may seem drastic, changes in the neural input-output curve can have a profound effect on the computations carried out by a neural network, and even enable substantial learning to occur on random networks [46].

### 4.3 Results

**Confirmation of the theory on a toy model.** Our theoretical results rely on assumptions such as strong convexity which may not hold in practice. Therefore, we first study the behavior of the contrastive meta-learning rule in a simple single-task regularization learning problem [7, 22–24, 27] using the synaptic model. Here, the goal of meta-learning is to determine meta-plasticity parameters  $\lambda$  in Eq. 7 (with  $\omega$  fixed at zero) which improve the generalization ability of the lower-level learning process. Formally, we wish to minimize the cross-validation objective  $L(\phi_\lambda^*, \lambda, \mathcal{D}^{\text{val}})$  for a single task, where  $\phi_\lambda^*$  is learned on a distinct set of data  $\mathcal{D}^{\text{train}}$  (cf. Eq. 5). We study a nonlinear neural network model with a small hidden layer (see Methods) so that the true meta-gradient can be easily computed and compared against. These findings are complemented by a detailed analysis of an analytically tractable quadratic model in Appendix E.

We measure the meta-gradient estimation error and control the fixed point approximation errors by varying the number of gradient descent steps. Theoretical guarantees on gradient descent ensure that, under some conditions on  $L^{\text{in}}$ ,  $-\log \delta$  is proportional to the number of steps taken. When the number of steps is the same in the two phases, we retrieve a very similar behavior to that predicted by our theory (Fig. 2, A and B).

We probe the  $(\delta, \delta')$  space in a different way, by fixing the total number of steps and then modifying the allocation across the two phases (Fig. 2C). The best achievable error, as a function of  $\beta$ , decreases before some  $\beta^*$  value and then increases, following the predictions of Theorem 2: too small  $\beta$  values turn out to hurt performance when the fixed points cannot be approximated arbitrarily well. Interestingly, the error plateaus for large  $\beta$  values and the size of the plateau decreases with  $\beta$  until reaching a critical value where it disappears. A conservative choice would therefore be to overestimate  $\beta$ , diminishing the meta-gradient estimation sensitivity to a sub-optimal allocation, with only a minor degradation in the best achievable quality.

When  $\beta$  approaches 0, the second phase optimization gets easier as the two fixed points become closer. Getting a better estimate of the first phase fixed point will be increasingly helpful for the second; the optimal fraction of steps allocated to the first phase hence decreases with  $\beta$ .

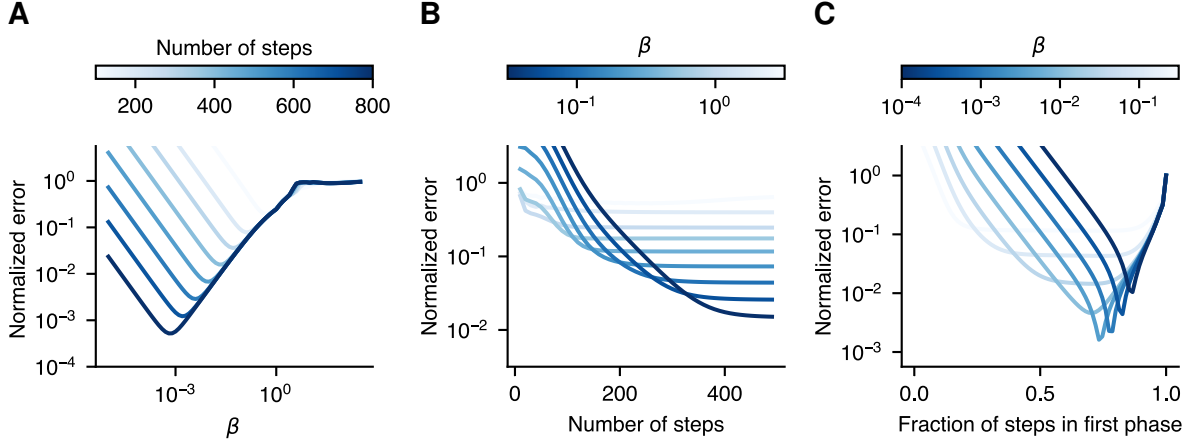


Figure 2: Confirmation of the theory on a toy model. (A, B) Evolution of the meta-gradient estimation error normalized by the norm of the meta-gradient, as a function of  $\beta$  and of the number of gradient descent steps performed in the two phases. (C) Evolution of the error with the fraction of steps allocated to the first phase, the total number of steps being fixed.

**High-dimensional regularization learning experiments.** We continue studying single-task regularization learning problems, now on the more complex vision datasets MNIST [47] and CIFAR-10 [48] using deep neural network models (see Methods). In particular, we compare our contrastive meta-learning rule to T1T2 [25], an implicit differentiation method that heuristically approximates the meta-gradient by a single Hessian-vector product, neglecting the inverse of the inner loss Hessian which arises when using implicit differentiation. The T1T2 meta-parameter update may or may not be considered biologically plausible, depending on whether the resulting Hessian-vector product is sufficiently simple to implement in a neural circuit. We focus on T1T2 for this reason, and because it is often a strong contender in practice [27].

A key strength of our method is that its meta-gradient estimate can be improved by reducing the lower-level fixed-point approximation error. By contrast, the bias in the T1T2 estimate cannot be reduced. In Fig. 3A we monitor the validation accuracy of both algorithms for a varying number of lower-level optimizer steps. As predicted, increasing the number of steps taken by the lower-level optimizer improves meta-learning performance for our algorithm whereas T1T2 eventually ceases to benefit from more steps. Notably, contrastive meta-learning is able to maintain a higher validation accuracy than T1T2 even for few steps. Table 1 shows extended results comparing the performance of the two algorithms for different choices of the training and validation set sizes. We focus on a low data regime, where the lower-level learner only has access to a very small training set, in the extreme case a single training example. While this regime does not necessarily yield the best results in terms of test set generalization, it increases the difficulty of the meta-learning problem. Our rule outperforms T1T2 in terms of both final validation accuracy and test accuracy throughout.

We repeat this analysis for contrastive meta-learning in Fig. 3B, while additionally varying the nudging

Table 1: Validation and test set accuracy (%) after regularization learning on MNIST and CIFAR-10 for different numbers of training samples and different splits of the training data into training and validation set. The split column shows the fraction of samples in the training set. Test accuracy is computed on the full, separate test set. All runs are repeated over 12 independent seeds and average accuracy as well as standard error of the mean are reported.

DATASET			CONTRASTIVE META-LEARNING		T1T2	
Name	Samples	Split	Validation set	Test set	Validation set	Test set
MNIST	60000	0.5	$99.75 \pm 0.06$	$98.04 \pm 0.04$	$98.64 \pm 0.05$	$97.57 \pm 0.03$
MNIST	10000	0.9999	$93.50 \pm 0.63$	$89.49 \pm 0.55$	$87.77 \pm 0.66$	$85.89 \pm 0.55$
MNIST	10000	0.999	$96.19 \pm 0.32$	$91.52 \pm 0.28$	$89.84 \pm 2.62$	$87.19 \pm 2.49$
CIFAR-10	1000	0.95	$99.77 \pm 0.13$	$31.68 \pm 0.26$	$62.97 \pm 3.73$	$28.69 \pm 0.85$

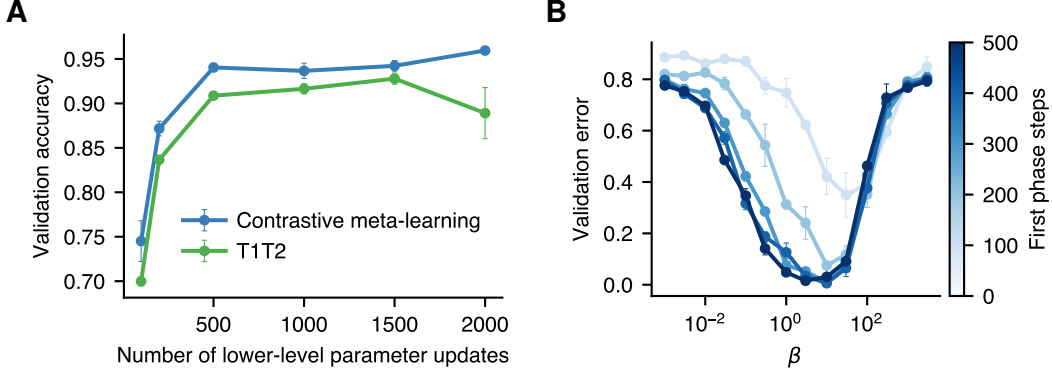


Figure 3: High-dimensional regularization learning experiments. (A) Validation accuracy on MNIST with 10 training samples and 9990 validation samples for varying number of lower-level parameter updates, comparing contrastive meta-learning and T1T2. (B) Validation error on CIFAR-10 as a function of  $\beta$  for 50 training samples and 950 validation samples for varying number of first-phase steps. Each data point is averaged over 12 independent seeds in (A) and 3 independent seeds in (B). Error bars denote the standard error of the mean.

strength  $\beta$ . In close correspondence to the behavior predicted by our theory (cf. Fig. S1 in Appendix C), the optimal value for  $\beta$  decreases as the number of first-phase steps increases, whereas the overall validation error of the optimal solution improves as the number of first-phase steps is increased.

**Few-shot learning.** We finally consider a few-shot learning problem and compare our contrastive meta-learning rule to the MAML method [10], which relies on backpropagation-through-training, as well as to its first-order approximation and the closely related Reptile algorithm [28]. We also include results obtained with implicit differentiation (iMAML [26]). Contrastive meta-learning lies between first-order approximate methods and iMAML. On the one hand, it only uses first-order derivatives, but it can approximate the meta-gradient with arbitrary precision. On the other hand, it requires the training process to be at equilibrium, but it does not resort to implicit differentiation and Hessian-vector products to estimate the meta-gradient.

Our results show that our rule performs better than the first-order methods considered here but not as well as iMAML, especially when this method is equipped with second-order optimization in its first phase (cf. Table 2). We simulate both the learning-by-modulation and synaptic models, the latter with fixed regularization strength  $\lambda$  and meta-learned slow weight  $\omega$  to compare directly to iMAML. We find that the synaptic and the learning-by-modulation models perform comparably, despite the fact that learning-by-modulation only requires adjusting a small number of parameters to learn new tasks.

Table 2: Few-shot learning of Omniglot characters. Results of related work are taken from the original papers, except for first-order MAML which is reported in ref. [28]. We report results obtained with contrastive meta-learning for the synaptic (syn.) and learning-by-modulation (mod.) models. We present test set classification accuracy (%) averaged over 5 seeds  $\pm$  std.

Method	5-way 1-shot	5-way 5-shot	20-way 1-shot	20-way 5-shot
MAML [10]	98.7 $\pm$ 0.4	99.9 $\pm$ 0.1	95.8 $\pm$ 0.3	98.9 $\pm$ 0.2
First-order MAML [10]	98.3 $\pm$ 0.5	99.2 $\pm$ 0.2	89.4 $\pm$ 0.5	97.9 $\pm$ 0.1
Reptile [28]	97.68 $\pm$ 0.04	99.48 $\pm$ 0.06	89.43 $\pm$ 0.14	97.12 $\pm$ 0.32
iMAML (GD) [26]	99.16 $\pm$ 0.35	99.67 $\pm$ 0.12	94.46 $\pm$ 0.42	98.69 $\pm$ 0.1
iMAML (Hessian-free) [26]	99.50 $\pm$ 0.26	99.74 $\pm$ 0.11	96.18 $\pm$ 0.36	99.14 $\pm$ 0.1
Contrastive meta-learning (syn.)	98.11 $\pm$ 0.34	99.49 $\pm$ 0.16	94.16 $\pm$ 0.12	98.06 $\pm$ 0.26
Contrastive meta-learning (mod.)	98.05 $\pm$ 0.06	99.45 $\pm$ 0.04	94.24 $\pm$ 0.39	98.60 $\pm$ 0.27



## 5 Discussion

We have introduced and analyzed contrastive meta-learning, a first-order method to learn meta-parameters. Unlike the conventional backpropagation-through-training procedure, which requires differentiable update rules, our method is agnostic to the choice of lower-level learning algorithm, as long as learning is treated as an optimization process. This opens the possibility of using combinatorial or stochastic search procedures to learn parameters. Contrastive meta-learning is also agnostic to how gradients with respect to meta-parameters are estimated. Thus, our meta-learning rule can be implemented in conjunction with any of the existing proposals for backpropagation in the brain. Likewise, the flexibility of our rule can simplify the implementation of meta-learning systems in specialized neuromorphic hardware, in particular on hardware where gradient-based learning is already available.

Being arguably as simple to implement as previous first-order methods [10, 25, 28], contrastive meta-learning allows reducing the meta-gradient approximation error by increasing the precision of the lower-level learning process. This brought measurable improvements in performance in our experiments comparing against the T1T2, Reptile and first-order MAML methods. The simplicity of our rule comes at the cost of requiring that the lower-level learning process is at equilibrium. We note that this does not mean that the learner must reach an expert level, since minima of the inner loss can correspond to high training error. For instance, in our few-shot learning experiments, performance on any given task is initially poor irrespective of the number of parameter updates taken by the learner, before letting meta-learning unfold over a sufficiently large number of tasks.

While we have only explored feedforward neural networks in the experiments presented here, the contrastive meta-learning rule can be readily applied to recurrent neural networks driven by time-varying input. It should be noted that the limitation to static input patterns that appears in the original application of equilibrium propagation does not apply at the meta-level.

Our theory describes how the meta-gradient estimation error depends on the precision of the lower-level learning process. In our upper bound, this error scales with the square root of the fixed point approximation errors, while a linear scaling can be provably achieved with implicit differentiation methods [24, 26]. This slower decrease in estimation error is the price to pay for the simplicity of the contrastive meta-learning rule.

Learning-by-modulation performed well in our experiments, despite the low-dimensionality of the parameters adapted to learn new tasks. This model is a departure from the orthodox view of synaptic plasticity as the sole basis for learning. The goal of synaptic plasticity is not to adjust synapses to solve any single task in isolation. Instead, synaptic plasticity is seen as a meta-learning process. This process ought to ensure that the neural activity produced in response to an input pattern provides an appropriate basis for fast learning of different tasks, each specified by a different pattern of firing rate modulation. As a family of tasks is mastered, the rate of synaptic plasticity naturally slows down and learning of new tasks mostly occurs by searching for the correct pattern of modulation. Such patterns of task-specific modulation could originate in higher-order brain areas, which can exert a strong top-down influence in how sensory inputs are processed [49]. Our view of synaptic weights as meta-parameters is reminiscent of a recent theory positing that the prefrontal cortex is best understood as a meta-learning system [50], in which synaptic weights are meta-parameters and the neural activity implements a learning algorithm.

Our meta-learning rule accumulates and contrasts gradients across phases. The learner is first presented with the original learning problem, subsequently followed by an additional problem that is artificially introduced by our algorithm. In the supervised meta-learning setting considered here, this artificial problem is obtained simply by augmenting the original dataset with held out data, with a rescaled importance determined by the nudging strength. Intuitively, contrastive meta-learning improves the generalization ability of a learner by concealing some data, that is later revealed to the learner. This process of buffering data, or experiences, which are later replayed to cortical networks is precisely the role ascribed to the hippocampus in the complementary learning systems theory [51]. Our work therefore adds meta-learning to the repertoire of functions for which replay might be critical [52]. In our algorithm, this systems process must act in concert with a synaptic or circuit mechanism, responsible for holding gradient information across phases. Investigating contrastive meta-learning with more detailed implementations of such mechanisms is an exciting direction for future work.

## Methods

**Toy regularization learning experiment.** We study a regularization learning problem [22–24] on the Boston housing dataset [53] (70% training and 30% validation split) using the synaptic model (Eq. 7), where  $\omega$  is taken to be 0. We simulate a small neural network comprising one hidden layer of 20 neurons with a hyperbolic tangent transfer function.

**MNIST and CIFAR-10 regularization learning experiments.** For training we choose a random subset of the training data and further split it into a training set and a validation set. This split is varied across different experiments to control the difficulty of the meta-learning problem. For MNIST [47] experiments we use a feedforward neural network with 5 hidden layers of size 256 and hyperbolic tangent nonlinearity. For CIFAR-10 [48] experiments we switch to a modified version of the classic LeNet-5 model [54] where we insert batch normalization layers [55] before each nonlinearity and replace the tanh nonlinearities with rectified linear units. For both MNIST and CIFAR-10 we normalize the inputs to have zero mean and unit variance but refrain from applying additional data augmentations. Our models are optimized using Adam [56] for outer-level optimization, and gradient descent with Nesterov momentum of 0.9 [57] for the inner-level optimization. We use minibatch stochastic gradient descent on the 60000-samples MNIST experiment, with a minibatch size of 500.

To choose the remaining hyperparameters of the meta-learning algorithms, we conduct grid searches (see Appendix F for search spaces). In cases where this grid is too high dimensional to be searched exhaustively, we sample randomly from the specified grid. In addition, we use a scheduler with a grace period of 10 meta-parameter steps to stop badly performing configurations early on.

**Few-shot learning experiments.** We follow the standard experimental setup for our Omniglot [14] experiments [10, 26, 58, 59] but use max-pooling instead of stride in the convolutional layers. We evaluate the statistics of batch normalization units [55] on the test set as in ref. [10], which yields a transductive classifier. Each task consists in learning to classify a new set of character types. The difficulty of the experiment is determined by the number of different classes (N-way) and the available data per class (K-shot) in the dataset. We investigate the performance of both the complex synapse and the learning-by-modulation meta-parameter models. Our objective is to minimize the expected classification error over tasks.

Hyperparameters that are specific to contrastive meta-learning are the following.

For all four few-shot setups, we use gradient descent with learning rate 0.01 and Nesterov momentum of 0.9 [57] as the lower-level optimizer. At the meta-level, we use Adam [56] with PyTorch default parameters (learning rate 0.001). To approximate the meta-gradient, we set  $\beta = 0.1$  and use the symmetric finite difference estimator, see Appendix D. We thus perform three phases, one for  $\beta = 0$ , one for  $\beta = -0.1$  and one for  $\beta = 0.1$ . Each phase has its unique optimization hyperparameters.

For proper convergence, we run the first phase ( $\beta = 0$ ) in all experiments for 200 gradient descent steps. Note that this duration was not tuned and could possibly be decreased to speed up experiments. We keep the first-phase ( $\beta = 0$ ) momentum terms of the optimizer for the two other ones.

For the complex synapse model (Eq. 7), we fix every  $\lambda_i$  to 0.1. We set the duration of the second (positive  $\beta$ ) and third phase (negative  $\beta$ ) to 100 gradient steps, as it yields the most robust results.

To implement our transfer function modulation model, we take advantage of the existence of batch normalization layers in our neural networks and consider the gain and shift parameters of these units as well as the parameters in the output layer as our parameters  $\phi$ . Here, the duration of the second and third phase is set to 50 gradient steps and we drop the regularization term.

For all experiments, we used the PyTorch default weight initialisation and trained the models on 3750 batches of size 16 for the 20-way and 32 for the 5-way setups.

**Code.** Deep learning experiments are implemented in PyTorch [60] and hyperparameter searches are conducted using Ray Tune [61]. Code is available upon request to the first authors.

## Acknowledgements

This work was supported by an Ambizione grant (PZ00P3186027) awarded to J.S. from the Swiss National Science Foundation and through funding from the Swiss Data Science Center (J.v.O. P18-03). We thank Angelika Steger, Benjamin Scellier, Giacomo Indiveri, Jean-Pascal Pfister, Mark van Rossum, Seijin Kobayashi and our colleagues at the Institute of Neuroinformatics for stimulating discussions and helpful comments.

## References

- [1] Jonathan Baxter. Theoretical models of learning to learn. In Sebastian Thrun and Lorien Pratt, editors, *Learning to learn*, pages 159–179. Springer US, 1998.
- [2] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Diploma thesis, Institut für Informatik, Technische Universität München, 1987.
- [3] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.
- [4] Jane X. Wang. Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences*, 38:90–95, 2021.
- [5] Johanni Brea and Wulfram Gerstner. Does computational neuroscience need new synaptic learning paradigms? *Current Opinion in Behavioral Sciences*, 11:61–66, 2016.
- [6] Tom Heskes. Solving a huge number of similar tasks: a combination of multi-task learning and a hierarchical Bayesian approach. In *International Conference on Machine Learning*, 1998.
- [7] David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.
- [8] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical Bayes. In *International Conference on Learning Representations*, 2018.
- [9] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019.
- [10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.
- [11] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016.
- [12] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, 2016.
- [13] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: a survey. *arXiv preprint arXiv:2004.05439*, 2020.
- [14] Brenden M Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2011.
- [15] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, 2019.
- [16] Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11, 2017.
- [17] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.
- [18] Carsten Peterson and James R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.
- [19] Javier R. Movellan. Contrastive Hebbian learning in the continuous Hopfield model. In *Connectionist Models*, pages 10–17. Elsevier, 1991.
- [20] Pierre Baldi and Fernando Pineda. Contrastive learning and neural oscillations. *Neural Computation*, 3(4):526–545, 1991.
- [21] Randall C. O’Reilly. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, 8(5):895–938, 1996.
- [22] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12(8):1889–1900, 2000.
- [23] Chuan-Sheng Foo, Chuong B. Do, and Andrew Y. Ng. Efficient multiple hyperparameter learning for log-linear models. In *Advances in Neural Information Processing Systems*, 2007.

- [24] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, 2016.
- [25] Jelena Luketina, Mathias Berglund, Klaus Greff, and Tapani Raiko. Scalable gradient-based tuning of continuous regularization hyperparameters. In *International Conference on Machine Learning*, 2016.
- [26] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, 2019.
- [27] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, 2020.
- [28] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, October 2018.
- [29] Adam H. Marblestone, Greg Wayne, and Konrad P. Kording. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience*, 10, 2016.
- [30] Timothy P. Lillicrap, Daniel Counden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1):13276, 2016.
- [31] Jordan Guerguiev, Timothy P. Lillicrap, and Blake A. Richards. Towards deep learning with segregated dendrites. *eLife*, 6:e22901, 2017.
- [32] James C. R. Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity. *Neural Computation*, 29(5):1229–1262, 2017.
- [33] Friedemann Zenke and Surya Ganguli. Superspike: supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018.
- [34] João Sacramento, Rui P. Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in Neural Information Processing Systems*, 2018.
- [35] Pieter R. Roelfsema and Anthony Holtmaat. Control of synaptic plasticity in deep cortical networks. *Nature Reviews Neuroscience*, 19(3):166–180, 2018.
- [36] Blake A. Richards and Timothy P. Lillicrap. Dendritic solutions to the credit assignment problem. *Current Opinion in Neurobiology*, 54:28–36, 2019.
- [37] James C. R. Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in Cognitive Sciences*, 23(3):235–250, 2019.
- [38] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11(1):3625, 2020.
- [39] Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.
- [40] Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *bioRxiv*, 2020.
- [41] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Meta-learning update rules for unsupervised representation learning. In *International Conference on Learning Representations*, 2019.
- [42] Marcus K. Benna and Stefano Fusi. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19(12):1697–1706, 2016.
- [43] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharmashan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 2017.
- [44] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017.
- [45] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702, May 2019.
- [46] Jake P. Stroud, Mason A. Porter, Guillaume Hennequin, and Tim P. Vogels. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature Neuroscience*, 21(12):1774, 2018.
- [47] Yann LeCun. The MNIST database of handwritten digits. Available at <http://yann.lecun.com/exdb/mnist>, 1998.
- [48] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

- [49] Matthew Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141–151, 2013.
- [50] Jane X. Wang, Zeb Kurth-Nelson, Dhharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6):860–868, 2018.
- [51] James L. McClelland, Bruce L. McNaughton, and Randall C. O’Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.
- [52] Tyler L. Hayes, Giri P. Krishnan, Maxim Bazhenov, Hava T. Siegelmann, Terrence J. Sejnowski, and Christopher Kanan. Replay in deep learning: current approaches and missing biological elements. *arXiv preprint arXiv:2104.04132*, 2021.
- [53] David Harrison Jr. and Daniel L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.
- [54] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [55] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.
- [56] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- [57] Y. Nesterov. A method for solving the convex programming problem with convergence rate  $O(1/k^2)$ . *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.
- [58] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 2016.
- [59] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.
- [60] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Workshop on Autodiff*, 2017.
- [61] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.
- [62] Benjamin Scellier. *A deep learning theory for neural networks grounded in physics*. PhD thesis, Université de Montréal, 2021.
- [63] Asen L. Dontchev and R. Tyrrell Rockafellar. *Implicit Functions and Solution Mappings*, volume 543. Springer, 2009.
- [64] Axel Laborieux, Maxence Ernoul, Benjamin Scellier, Yoshua Bengio, Julie Grollier, and Damien Querlioz. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in Neuroscience*, 14, 2021.
- [65] H.Z. Hassan, A.A. Mohamad, and G.E. Atteia. An algorithm for the finite difference approximation of derivatives with arbitrary degree and order of accuracy. *Journal of Computational and Applied Mathematics*, 236(10):2622–2631, 2012.

## A Equilibrium propagation

We review a more precise statement of the equilibrium propagation result (Theorem 1) as presented in [62] and restate the assumptions needed for that result. Recall the augmented loss

$$\mathcal{L}(\phi, \theta, \beta) = L^{\text{in}}(\phi, \theta) + \beta L^{\text{out}}(\phi, \theta).$$

**Theorem 3.** *Let  $L^{\text{in}}$  and  $L^{\text{out}}$  be two twice continuously differentiable functions. Let  $\phi^*$  be a fixed point of  $\mathcal{L}(\cdot, \bar{\theta}, \bar{\beta})$ , i.e.*

$$\frac{\partial \mathcal{L}}{\partial \phi}(\phi^*, \bar{\theta}, \bar{\beta}) = 0,$$

*such that  $\partial_{\phi}^2 \mathcal{L}(\phi^*, \bar{\theta}, \bar{\beta})$  is invertible. Then, there exists a neighborhood of  $(\bar{\theta}, \bar{\beta})$  and a continuously differentiable function  $(\theta, \beta) \mapsto \phi_{\theta, \beta}^*$  such that  $\phi_{\bar{\theta}, \bar{\beta}}^* = \phi^*$  and for every  $(\theta, \beta)$  in this neighborhood*

$$\frac{\partial \mathcal{L}}{\partial \phi}(\phi_{\theta, \beta}^*, \theta, \beta) = 0.$$

Furthermore,

$$\frac{d}{d\theta} \frac{\partial \mathcal{L}}{\partial \beta}(\phi_{\theta, \beta}^*, \theta) = \frac{d}{d\beta} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_{\theta, \beta}^*, \theta, \beta).$$

*Proof.* The first point follows from the implicit function theorem [63]. Let  $(\theta, \beta)$  be in a neighborhood of  $(\bar{\theta}, \bar{\beta})$  in which  $\phi_{\theta, \beta}^*$  is differentiable.

The symmetry of second order derivatives of a scalar function implies that

$$\frac{d}{d\theta} \frac{d}{d\beta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta) = \frac{d}{d\beta} \frac{d}{d\theta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta).$$

We then simplify the two sides of the equation. First, we look at the left-hand side and simplify  $d_{\beta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta)$  using the chain rule and the fixed point condition:

$$\begin{aligned} \frac{d}{d\beta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta) &= \frac{\partial \mathcal{L}}{\partial \beta}(\phi_{\theta, \beta}^*, \theta, \beta) + \frac{\partial \mathcal{L}}{\partial \phi}(\phi_{\theta, \beta}^*, \theta, \beta) \frac{d\phi_{\theta, \beta}^*}{d\beta} \\ &= \frac{\partial \mathcal{L}}{\partial \beta}(\phi_{\theta, \beta}^*, \theta, \beta). \end{aligned}$$

Similarly, the  $d_{\theta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta)$  term in the right-hand side is equal to  $\partial_{\theta} \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta)$  and we have the required result.  $\square$

Note that we only need the invertible Hessian condition to define an implicit function  $\phi_{\theta, \beta}^*$ . This is made possible by the implicit function theorem. However, we do not need the implicit differentiation formula given by the theorem, in contrast to implicit methods such as iMAML [26].

## B On the locality of the contrastive meta-learning rule

Depending on the problem formulation, the contrastive meta-learning rule

$$\hat{\nabla}_{\theta} = \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_{\beta}, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_0, \theta, 0) \right)$$

can be entirely local, in the sense that it only depends on the parameters within the same unit (e.g. a synapse or a neuron).

**Complex synapse model.** We consider the general setting in which

$$L^{\text{in}}(\phi, \theta) = f(\phi) + \frac{1}{2} \sum_i \lambda_i (\phi_i - \omega_i)^2$$

$$L^{\text{out}}(\phi) = g(\phi)$$

where  $f$  and  $g$  are two losses (e.g. training and validation losses) that only depend on  $\phi$ ,  $\lambda$  is the regularization strength and  $\omega$  is the regularization center. Note that the formulation above incorporates the regularization learning ( $\omega$  fixed to 0) and initialization learning (each  $\lambda_i$  is fixed to some scalar value  $\lambda$ ) settings, which we have considered in our experiments. Then,

$$\frac{\partial \mathcal{L}}{\partial \lambda_i}(\phi, \theta, \beta) = \frac{1}{2}(\phi_i - \omega_i)^2$$

$$\frac{\partial \mathcal{L}}{\partial \omega_i}(\phi, \theta, \beta) = -\lambda_i(\phi_i - \omega_i).$$

The locality of the learning rule follows, as the updates of the parameters of the unit  $i$  only depend on  $\lambda_i$ ,  $\omega_i$  and the value of  $\phi_i$  at the solutions of the two phases.

## C Proof of the theoretical results

Recall that we estimate the meta-gradient

$$\nabla_\theta = \frac{d}{d\theta} L^{\text{out}}(\phi_\theta^*, \theta)$$

with the equilibrium propagation estimate

$$\widehat{\nabla}_\theta = \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_\beta, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_0, \theta, 0) \right)$$

for  $\hat{\phi}_0$  and  $\hat{\phi}_\beta$  two approximations of the inner optimization problem solutions  $\phi_0^*$  and  $\phi_\beta^*$ . In addition to those two quantities, we introduce

$$\widehat{\nabla}_\theta^* := \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\phi_{\theta, \beta}^*, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\phi_{\theta, 0}^*, \theta, 0) \right),$$

the equilibrium propagation estimate evaluated at the exact fixed points.

The introduction of  $\widehat{\nabla}_\theta^*$  helps to distinguish the two types of error introduced in the estimation of  $\nabla_\theta$ . Those errors consist of:

- the finite difference error  $\|\nabla_\theta - \widehat{\nabla}_\theta^*\|$ , namely the error stemming from the approximation of  $d_\beta \partial_\theta \mathcal{L}(\phi_\beta^*, \theta, \beta)|_{\beta=0}$  with a finite difference.
- the fixed point approximation induced error  $\|\widehat{\nabla}_\theta^* - \widehat{\nabla}_\theta\|$  that is the consequence of the approximation of the fixed points.

Under some assumptions (Assumption 1), the two errors can be upper bounded, yielding Theorem 2.

**Assumption 1.** Assume that  $L^{\text{in}}$  and  $L^{\text{out}}$  are three-times continuously differentiable and that they, as functions of  $\phi$ , verify the following properties.

- i.  $\partial_\theta L^{\text{in}}$  is  $B^{\text{in}}$ -Lipschitz and  $\partial_\theta L^{\text{out}}$  is  $B^{\text{out}}$ -Lipschitz.
- ii.  $L^{\text{in}}$  and  $L^{\text{out}}$  are  $L$ -smooth and  $\mu$ -strongly convex.
- iii. their Hessians are  $\rho$ -Lipschitz.
- iv.  $\partial_\phi \partial_\theta L^{\text{in}}$  and  $\partial_\phi \partial_\theta L^{\text{out}}$  are  $\sigma$ -Lipschitz.

**Theorem 2.** Let  $\beta > 0$  and  $(\delta, \delta')$  be such that

$$\|\phi_{\theta,0}^* - \hat{\phi}_0\| \leq \delta$$

and

$$\|\phi_{\theta,\beta}^* - \hat{\phi}_\beta\| \leq \delta'.$$

Under Assumption 1, there exists a  $\theta$ -dependent constant  $C$  such that

$$\|\nabla_\theta - \hat{\nabla}_\theta\| \leq \frac{B^{\text{in}}(\delta + \delta')}{\beta} + B^{\text{out}}\delta' + C \frac{\beta}{1 + \beta} =: \mathcal{B}(\delta, \delta', \beta).$$

If we additionally assume that  $\theta$  lies in a compact set, we can choose  $C$  to be independent of  $\theta$ .

We provide an additional visualization of this bound to the ones of Fig. 1 in Fig. S1, this time when  $\delta'$  is fixed to some given value.

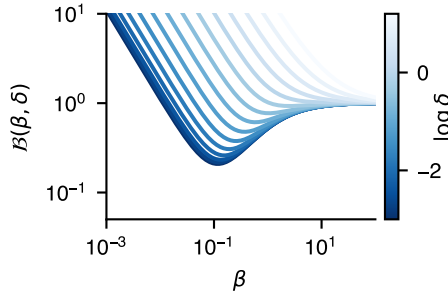


Figure S1: Visualization of the bound  $\mathcal{B}$  from Theorem 2 ( $C = 1$ ,  $B^{\text{in}} = B^{\text{out}} = 1$ ), as a function of  $\beta$ , when  $\delta'$  is fixed to  $10^{-2}$ .

We prove this result in two steps: we first introduce the technical lemmas needed for the proof and the proof itself in Appendix C.1 and then prove those technical lemmas in Appendix C.2. In the following, for conciseness, we omit  $\theta$  in the notations when it is considered fixed.

## C.1 Proof of Theorem 2

The idea behind the proof of Theorem 2 is the following: we separate the meta-gradient approximation error  $\|\nabla_\theta - \hat{\nabla}_\theta\|$  into the finite difference error  $\|\nabla_\theta - \hat{\nabla}_\theta^*\|$  and the fixed point approximation induced error  $\|\hat{\nabla}_\theta^* - \hat{\nabla}_\theta\|$  and then individually bound those two errors.

Bounding the second term directly results from the Lipschitz continuity property of the partial derivatives (Assumption 1.i). It yields the  $B^{\text{in}}(\delta + \delta')/\beta + B^{\text{out}}\delta'$  part of the bound.

The first term requires more work. We use Taylor's Theorem to show that  $\hat{\nabla}_\theta^* - \nabla_\theta$  is equal to some integral remainder. It then remains to bound what is inside the integral remainder, which is the second order derivative  $d_\beta^2 \partial_\theta \mathcal{L}(\phi_\beta^*, \beta)$ . This is done in the Lemmas presented in this section: Lemma 1 allows us to get uniform bounds, Lemmas 2 and 3 bound the first and second order derivatives of  $\beta \mapsto \phi_\beta^*$  and Lemma 4 bounds  $d_\beta^2 \partial_\theta \mathcal{L}(\phi_\beta^*, \beta)$  with the norm of the two derivatives we have just bounded. The proofs of those four lemmas being mainly technical, we present them in a separate section (Appendix C.2).

**Lemma 1.** Under Assumption 1.ii, if  $\theta$  lies in a compact set  $\mathcal{D}$  the function  $(\theta, \beta) \mapsto \phi_{\theta,\beta}^*$  is uniformly bounded.

**Lemma 2.** Under Assumption 1.ii, there exists a  $\theta$ -dependent constant  $R$  s.t., for every positive  $\beta$ ,

$$\left\| \frac{d\phi_\beta^*}{d\beta} \right\| \leq \frac{LR}{(1 + \beta)^2 \mu}.$$

If we additionally assume that  $\theta$  lies in a compact set, we can choose  $R$  to be independent of  $\theta$ .



**Remark 1.** A side product of the proof of Lemma 2 (see Appendix C.2.3 for a proof) is a bound on the distance between the minimizer of  $\mathcal{L}$  and the minimizers of  $L^{\text{in}}$  and  $L^{\text{out}}$ . We have

$$\|\phi_\beta^* - \phi_\infty^*\| \leq \frac{1}{1 + \beta}$$

and

$$\|\phi_\beta^* - \phi_0^*\| \leq \frac{\beta}{1 + \beta}$$

up to some constant factors.

**Lemma 3.** Under Assumptions 1.ii and 1.iii,

$$\left\| \frac{d^2 \phi_\beta^*}{d\beta^2} \right\| \leq \frac{\rho}{\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|^2 + \frac{2L}{(1 + \beta)\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|.$$

When Lemma 3 is combined with Lemma 2,

$$\left\| \frac{d^2 \phi_\beta^*}{d\beta^2} \right\| \leq \frac{1}{(1 + \beta)^3}.$$

up to some constant factor.

**Lemma 4.** Under Assumptions 1.ii, 1.iii and 1.iv, there exists a constant  $M$  such that

$$\left\| \frac{d^2}{d\beta^2} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_\beta^*, \beta) \right\| \leq M \left( \left\| \frac{d\phi_\beta^*}{d\beta} \right\| + (1 + \beta) \left( \left\| \frac{d\phi_\beta^*}{d\beta} \right\|^2 + \left\| \frac{d^2 \phi_\beta^*}{d\beta^2} \right\| \right) \right).$$

We now have all the necessary tools to prove Theorem 2.

*Proof of Theorem 2.* We separate the sources of error within the meta-gradient estimation error using the triangle inequality:

$$\|\widehat{\nabla}_\theta - \nabla_\theta\| \leq \underbrace{\|\widehat{\nabla}_\theta - \widehat{\nabla}_\theta^*\|}_{a)} + \underbrace{\|\widehat{\nabla}_\theta^* - \nabla_\theta\|}_{b)}$$

We bound each of the error terms individually:

a) Recall that

$$\widehat{\nabla}_\theta = \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_\beta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_0, 0) \right)$$

and that a similar formula holds for  $\widehat{\nabla}_\theta^*$  (evaluated at the fixed points instead of the approximations). It follows

$$\|\widehat{\nabla}_\theta - \widehat{\nabla}_\theta^*\| \leq \frac{1}{\beta} \left( \left\| \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_\beta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\phi_\beta^*, \beta) \right\| + \left\| \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_0, 0) - \frac{\partial \mathcal{L}}{\partial \theta}(\phi_0^*, 0) \right\| \right).$$

Since  $\phi \mapsto \partial_\theta \mathcal{L}(\phi, \beta)$  is a  $(B^{\text{in}} + \beta B^{\text{out}})$ -Lipschitz function as a sum of  $\partial_\theta L^{\text{in}}$  and  $\partial_\theta L^{\text{out}}$ , two Lipschitz continuous functions with constants  $B^{\text{in}}$  and  $B^{\text{out}}$ ,

$$\begin{aligned} \|\widehat{\nabla}_\theta - \widehat{\nabla}_\theta^*\| &\leq \frac{B^{\text{in}} + \beta B^{\text{out}}}{\beta} \|\hat{\phi}_\beta - \phi_\beta^*\| + \frac{B^{\text{in}}}{\beta} \|\hat{\phi}_0 - \phi_0^*\| \\ &\leq \frac{B^{\text{in}} + \beta B^{\text{out}}}{\beta} \delta' + \frac{B^{\text{in}}}{\beta} \delta. \end{aligned}$$

b) Taylor's Theorem applied to  $\beta \mapsto \partial_\theta \mathcal{L}(\phi_\beta^*, \beta)$  up to the first order of differentiation yields

$$\frac{\partial \mathcal{L}}{\partial \theta}(\phi_\beta^*, \beta) = \frac{\partial \mathcal{L}}{\partial \theta}(\phi_0^*, 0) + \beta \frac{d}{d\beta} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_0^*, 0) + \int_0^\beta (\beta - t) \frac{d^2}{d\beta^2} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_t^*, t) dt.$$

The equilibrium propagation theorem (Theorem 3), which is applicable thanks to Assumption 1.ii, gives

$$\nabla_\theta = \frac{d}{d\beta} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_0^*, 0),$$

hence

$$\|\widehat{\nabla}_\theta^* - \nabla_\theta\| = \left\| \int_0^\beta (\beta - t) \frac{d^2}{d\beta^2} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_t^*, t) dt \right\|.$$

Using the integral version of Cauchy-Schwartz inequality, we have

$$\|\widehat{\nabla}_\theta^* - \nabla_\theta\| \leq \int_0^\beta (\beta - t) \left\| \frac{d^2}{d\beta^2} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_t^*, t) \right\| dt.$$

We now use Lemma 4 combined with Lemmas 2 and 3 to bound  $d_\beta^2 \partial_\theta \mathcal{L}(\phi_t^*, t)$ . We focus on the  $\beta$  dependencies and don't write any of the constant factors:

$$\begin{aligned} \left\| \frac{d^2}{d\beta^2} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_t^*, t) \right\| &\leq \left\| \frac{d\phi_t^*}{d\beta} \right\| + (1+t) \left( \left\| \frac{d\phi_t^*}{d\beta} \right\|^2 + \left\| \frac{d^2 \phi_t^*}{d\beta^2} \right\| \right) \\ &\leq \frac{1}{(1+t)^2} + (1+t) \left( \frac{1}{(1+t)^3} + \frac{1}{(1+t)^4} \right) \\ &\leq (1+t)^{-2}. \end{aligned}$$

It follows that

$$\begin{aligned} \|\widehat{\nabla}_\theta^* - \nabla_\theta\| &\leq \int_0^\beta \frac{(\beta - t)}{(1+t)^2} dt \\ &= (1+\beta) \int_0^\beta \frac{1}{(1+t)^2} dt - \int_0^\beta \frac{1}{(1+t)} dt \\ &= (1+\beta) \frac{\beta}{1+\beta} - \ln(1+\beta) \\ &\leq \beta - \frac{\beta}{1+\beta} \\ &= \frac{\beta^2}{1+\beta}. \end{aligned}$$

where the inequality comes from the well-known  $\ln(x) \geq 1 - \frac{1}{x}$  inequality for positive  $x$  (applied to  $x = 1 + \beta$ ). There hence exists a constant  $C$  such that

$$\|\widehat{\nabla}_\theta^* - \nabla_\theta\| \leq C \frac{\beta}{1+\beta}.$$

If  $\theta$  lies in a compact set, the bound in Lemma 2 is uniform over  $\theta$ . This is the only constant factor that depends on  $\theta$ , so the bound is uniform.  $\square$

## C.2 Proof of technical lemmas

In this section we prove the four technical lemmas that we need for Theorem 2.

### C.2.1 Proof of Lemma 1

**Lemma 1.** *Under Assumption 1.ii, if  $\theta$  lies in a compact set  $\mathcal{D}$  the function  $(\theta, \beta) \mapsto \phi_{\theta, \beta}^*$  is uniformly bounded.*

*Proof.* Let  $\alpha \in [0, 1]$ . Define

$$\mathcal{L}'(\phi, \theta, \alpha) := (1 - \alpha)L^{\text{in}}(\phi, \theta) + \alpha L^{\text{out}}(\phi, \theta).$$

As  $L^{\text{in}}$  and  $L^{\text{out}}$  are strongly-convex, there exists a unique minimizer  $\phi_{\theta,\alpha}^*$  of  $\phi \mapsto \mathcal{L}'(\phi, \theta, \alpha)$ . The implicit function theorem ensures that the function  $(\theta, \alpha) \mapsto \phi_{\theta,\alpha}^*$ , defined on  $\mathcal{D} \times [0, 1]$ , is continuous. As  $\mathcal{D} \times [0, 1]$  is a compact set,  $\phi_{\theta,\alpha}^*$  is then uniformly bounded. Now, remark that

$$\mathcal{L}(\phi, \theta, \beta) = (1 + \beta)\mathcal{L}'\left(\phi, \theta, \frac{\beta}{1 + \beta}\right)$$

and thus  $\phi_{\theta,\beta}^* = \phi_{\theta,\beta/(1+\beta)}^*$ . It follows that  $\phi_{\theta,\beta}^*$  is uniformly bounded.  $\square$

### C.2.2 Proof of Lemma 2

**Lemma 2.** *Under Assumption 1.ii, there exists a  $\theta$ -dependent constant  $R$  s.t., for every positive  $\beta$ ,*

$$\left\| \frac{d\phi_\beta^*}{d\beta} \right\| \leq \frac{LR}{(1 + \beta)^2\mu}.$$

*If we additionally assume that  $\theta$  lies in a compact set, we can choose  $R$  to be independent of  $\theta$ .*

*Proof.* The function  $\phi \mapsto \mathcal{L}(\phi, \beta)$  is  $(1 + \beta)\mu$ -strongly convex so its Hessian  $\partial_\phi^2 \mathcal{L}$  is invertible and its inverse has a spectral norm upper bounded by  $1/((1 + \beta)\mu)$ . The use of the implicit function theorem follows and gives

$$\begin{aligned} \left\| \frac{d\phi_\beta^*}{d\beta} \right\| &= \left\| -(\partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta))^{-1} \partial_\beta \partial_\phi \mathcal{L}(\phi_\beta^*) \right\| \\ &= \left\| -(\partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta))^{-1} \partial_\phi L^{\text{out}}(\phi_\beta^*) \right\| \\ &\leq \frac{1}{(1 + \beta)\mu} \|\partial_\phi L^{\text{out}}(\phi_\beta^*)\|. \end{aligned}$$

It remains to bound the gradient of  $L^{\text{out}}$ . Since  $\beta \mapsto \phi_\beta^*$  is continuous and has finite limits in 0 and  $\infty$  (namely the minimizers of  $L^{\text{in}}$  and  $L^{\text{out}}$ ), it evolves in a bounded set. There hence exists a positive constant  $R$  such that, for all positive  $\beta$ ,

$$\max(\|\phi_\beta^* - \phi_0^*\|, \|\phi_\beta^* - \phi_\infty^*\|) \leq \frac{R}{2}.$$

If  $\theta$  lies in a compact set, Lemma 1 guarantees that there exists such a constant that doesn't depend on the choice of  $\theta$ . We then bound the gradient of  $L^{\text{out}}$  using the smoothness properties of  $L^{\text{in}}$  and  $L^{\text{out}}$ , either directly

$$\|\partial_\phi L^{\text{out}}(\phi_\beta^*)\| \leq L\|\phi_\beta^* - \phi_\infty^*\| \leq \frac{LR}{2}$$

or indirectly, using the fixed point condition  $\partial_\phi \mathcal{L}(\phi_\beta^*, \beta) = 0$ ,

$$\|\partial_\phi L^{\text{out}}(\phi_\beta^*)\| = \frac{1}{\beta} \|\partial_\phi L^{\text{in}}(\phi_\beta^*)\| \leq \frac{L\|\phi_\beta^* - \phi_0^*\|}{\beta} \leq \frac{LR}{2\beta}.$$

The required result is finally obtained by remarking

$$\|\partial_\phi L^{\text{out}}(\phi_\beta^*)\| \leq \min\left(1, \frac{1}{\beta}\right) \frac{LR}{2} \leq \frac{LR}{1 + \beta}.$$

$\square$

### C.2.3 Proof of Remark 1

We now prove Remark 1, which directly follows from the previous proof. Recall that we have just proved

$$\|\partial_\phi L^{\text{out}}(\phi_\beta^*)\| \leq \frac{LR}{1 + \beta}.$$

With the strong convexity of  $L^{\text{out}}$ , the gradient is also lower bounded

$$\|\partial_\phi L^{\text{out}}(\phi_\beta^*)\| \geq \mu \|\phi_\beta^* - \phi_\infty^*\|,$$

meaning that

$$\|\phi_\beta^* - \phi_\infty^*\| \leq \frac{LR}{\mu(1+\beta)}.$$

Similarly, one can show that

$$\|\phi_0^* - \phi_\beta^*\| \leq \frac{\beta}{1+\beta}$$

up to some constant factor. This can be proved with

$$\|\phi_0^* - \phi_\beta^*\| \leq \frac{\|\partial_\phi L^{\text{in}}(\phi_\beta^*)\|}{\mu} = \frac{\beta \|\partial_\phi L^{\text{out}}(\phi_\beta^*)\|}{\mu} \leq \frac{\beta LR}{(1+\beta)\mu}.$$

### C.2.4 Proof of Lemma 3

**Lemma 3.** *Under Assumptions 1.ii and 1.iii,*

$$\left\| \frac{d^2 \phi_\beta^*}{d\beta^2} \right\| \leq \frac{\rho}{\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|^2 + \frac{2L}{(1+\beta)\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|.$$

*Proof.* The starting point of the proof is the implicit function theorem, that we differentiate with respect to  $\beta$  as a product of functions

$$\begin{aligned} \frac{d^2 \phi_\beta^*}{d\beta^2} &= \frac{d}{d\beta} \left( -(\partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta))^{-1} \partial_\phi L^{\text{out}}(\phi_\beta^*) \right) \\ &= - \underbrace{\left( \frac{d}{d\beta} \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta)^{-1} \right) \partial_\phi L^{\text{out}}(\phi_\beta^*)}_{a)} - \underbrace{\partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta)^{-1} \left( \frac{d}{d\beta} \partial_\phi L^{\text{out}}(\phi_\beta^*) \right)}_{b)}. \end{aligned}$$

We now individually calculate and bound each term.

a) The differentiation of the inverse of a matrix gives

$$a) = -\partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta)^{-1} \left( \frac{d}{d\beta} \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta) \right) \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta)^{-1} \partial_\phi L^{\text{out}}(\phi_\beta^*),$$

which we can rewrite as

$$a) = \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta)^{-1} \left( \frac{d}{d\beta} \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta) \right) \frac{d\phi_\beta^*}{d\beta}.$$

The derivative term in the middle of the r.h.s. is equal to

$$\begin{aligned} \frac{d}{d\beta} \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta) &= \frac{d}{d\beta} [\partial_\phi^2 L^{\text{in}}(\phi_\beta^*) + \beta \partial_\phi^2 L^{\text{out}}(\phi_\beta^*)] \\ &= \frac{d}{d\beta} \partial_\phi^2 L^{\text{in}}(\phi_\beta^*) + \beta \frac{d}{d\beta} \partial_\phi^2 L^{\text{out}}(\phi_\beta^*) + \partial_\phi^2 L^{\text{out}}(\phi_\beta^*). \end{aligned}$$

Using the Lipschitz continuity of the Hessians,

$$\left\| \frac{d}{d\beta} \partial_\phi^2 L^{\text{in}}(\phi_\beta^*) + \beta \frac{d}{d\beta} \partial_\phi^2 L^{\text{out}}(\phi_\beta^*) \right\| \leq (1+\beta)\rho \left\| \frac{d\phi_\beta^*}{d\beta} \right\|.$$

Adding the upper bound on the norm the Hessian of  $L^{\text{out}}$  due to its smoothness,

$$\left\| \frac{d}{d\beta} \partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta) \right\| \leq (1+\beta)\rho \left\| \frac{d\phi_\beta^*}{d\beta} \right\| + L.$$

We finally have

$$\begin{aligned}\|a)\| &\leq \frac{1}{\mu(1+\beta)} \left( (1+\beta)\rho \left\| \frac{d\phi_\beta^*}{d\beta} \right\| + L \right) \left\| \frac{d\phi_\beta^*}{d\beta} \right\| \\ &\leq \frac{\rho}{\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|^2 + \frac{L}{(1+\beta)\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|.\end{aligned}$$

b) With the chain rule,

$$\frac{d}{d\beta} \partial_\phi L^{\text{out}}(\phi_\beta^*) = \partial_\phi^2 L^{\text{out}}(\phi_\beta^*) \frac{d\phi_\beta^*}{d\beta}$$

so

$$\begin{aligned}\|b)\| &\leq \|\partial_\phi^2 \mathcal{L}(\phi_\beta^*, \beta)^{-1}\| \|\partial_\phi^2 L^{\text{out}}(\phi_\beta^*)\| \left\| \frac{d\phi_\beta^*}{d\beta} \right\| \\ &\leq \frac{L}{(1+\beta)\mu} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|.\end{aligned}$$

□

### C.2.5 Proof of Lemma 4

**Lemma 4.** *Under Assumptions 1.ii, 1.iii and 1.iv, there exists a constant  $M$  such that*

$$\left\| \frac{d^2}{d\beta^2} \frac{\partial \mathcal{L}}{\partial \theta}(\phi_\beta^*, \beta) \right\| \leq M \left( \left\| \frac{d\phi_\beta^*}{d\beta} \right\| + (1+\beta) \left( \left\| \frac{d\phi_\beta^*}{d\beta} \right\|^2 + \left\| \frac{d^2 \phi_\beta^*}{d\beta^2} \right\| \right) \right).$$

*Proof.* We want to bound the norm of  $\frac{d^2}{d\beta^2} \partial_\theta \mathcal{L}(\phi_\beta^*, \beta)$ . The first order derivative can be calculated with the chain rule of differentiation

$$\frac{d}{d\beta} \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) = \partial_\beta \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) + \partial_\phi \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) \frac{d\phi_\beta^*}{d\beta}.$$

We then once again differentiate this equation with respect to  $\beta$ . The  $\partial_\beta \partial_\theta \mathcal{L}(\phi_\beta^*, \beta)$  term has in fact, due to the nature of  $\mathcal{L}$ , no direct dependence on  $\beta$  and is equal to  $\partial_\theta L^{\text{out}}(\phi_\beta^*)$ . Hence

$$\frac{d}{d\beta} \partial_\beta \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) = \partial_\phi \partial_\theta L^{\text{out}}(\phi_\beta^*) \frac{d\phi_\beta^*}{d\beta}.$$

Differentiating the other term yields

$$\frac{d}{d\beta} \left[ \partial_\phi \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) \frac{d\phi_\beta^*}{d\beta} \right] = \left[ \partial_\beta \partial_\phi \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) + \partial_\phi^2 \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) \otimes \frac{d\phi_\beta^*}{d\beta} \right] \frac{d\phi_\beta^*}{d\beta} + \partial_\phi \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) \frac{d^2 \phi_\beta^*}{d\beta^2}.$$

Therefore,

$$\frac{d^2}{d\beta^2} \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) = 2\partial_\phi \partial_\theta L^{\text{out}}(\phi_\beta^*) \frac{d\phi_\beta^*}{d\beta} + \partial_\phi^2 \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) \otimes \frac{d\phi_\beta^*}{d\beta} \otimes \frac{d\phi_\beta^*}{d\beta} + \partial_\phi \partial_\theta \mathcal{L}(\phi_\beta^*, \beta) \frac{d^2 \phi_\beta^*}{d\beta^2}.$$

We now individually bound each term:

- due to Assumption 1.i,  $\phi \mapsto \partial_\theta L^{\text{out}}(\phi)$  is  $B^{\text{out}}$ -Lipschitz continuous, so  $\|\partial_\phi \partial_\theta L^{\text{out}}\| \leq B^{\text{out}}$  and

$$\left\| 2\partial_\phi \partial_\theta L^{\text{out}}(\phi_\beta^*) \frac{d\phi_\beta^*}{d\beta} \right\| \leq 2B^{\text{out}} \left\| \frac{d\phi_\beta^*}{d\beta} \right\|.$$

- similarly to the previous point,

$$\left\| \partial_\phi \partial_\theta \mathcal{L}(\phi_\beta^*) \frac{d^2 \phi_\beta^*}{d\beta^2} \right\| \leq (B^{\text{in}} + \beta B^{\text{out}}) \left\| \frac{d^2 \phi_\beta^*}{d\beta^2} \right\|.$$

– Assumption 1.iv ensures that  $\phi \mapsto \partial_\phi \partial_\theta \mathcal{L}(\phi, \beta)$  is  $(1 + \beta)\sigma$ -Lipschitz continuous and

$$\left\| \frac{\partial^3 \mathcal{L}}{\partial \phi^2 \partial \theta}(\phi_\beta^*, \beta) \otimes \frac{d\phi_\beta^*}{d\beta} \otimes \frac{d\phi_\beta^*}{d\beta} \right\| \leq (1 + \beta)\sigma \left\| \frac{d\phi_\beta^*}{d\beta} \right\|^2.$$

Take  $M := \max(2B^{\text{out}}, B^{\text{in}}, \sigma)$ : we now have the desired result.  $\square$

### C.3 Proof of Corollary 1

Recall the following Corollary.

**Corollary 1.** *Under Assumption 1, if we suppose that for every strictly positive  $\beta$  we approximate the two fixed points with precision  $\delta$  and  $\delta'$  and if  $(\delta + \delta') < C/B^{\text{in}}$ , the best achievable bound in Theorem 2 is smaller than*

$$B^{\text{out}}\delta' + 2\sqrt{CB^{\text{in}}(\delta + \delta')}$$

and is attained for  $\beta$  equal to

$$\beta^*(\delta, \delta') = \frac{\sqrt{B^{\text{in}}(\delta + \delta')}}{\sqrt{C} - \sqrt{B^{\text{in}}(\delta + \delta')}}.$$

We visualize the results from Corollary 1 on Fig. S2.

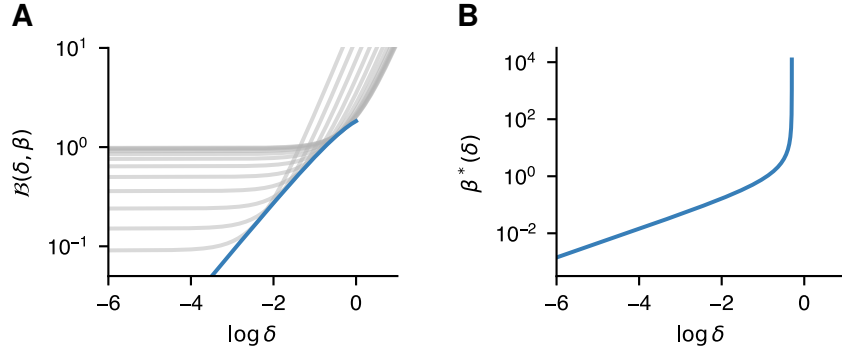


Figure S2: Visualization of Corollary 1. (A) Best achievable bound as a function of  $\delta = \delta'$  in blue<sup>2</sup>. The grey lines are the bounds from Theorem 2 we laid out on Fig. 1B. (B)  $\beta$  value that minimizes the bound, as a function of  $\delta = \delta'$ .

*Proof.* The  $\beta$  derivative of the bound  $\mathcal{B}$  obtained in Theorem 2 is

$$\frac{\partial \mathcal{B}}{\partial \beta}(\delta, \delta', \beta) = -\frac{B^{\text{in}}(\delta + \delta')}{\beta^2} + \frac{C}{(1 + \beta)^2}$$

and vanishes for  $\beta$  verifying

$$\beta \left( \sqrt{C} - \sqrt{B^{\text{in}}(\delta + \delta')} \right) = \sqrt{B^{\text{in}}(\delta + \delta')}.$$

As  $(\delta + \delta') < C/B^{\text{in}}$ , the previous criterion is met when  $\beta$  is equal to the positive

$$\beta^* := \frac{\sqrt{B^{\text{in}}(\delta + \delta')}}{\sqrt{C} - \sqrt{B^{\text{in}}(\delta + \delta')}}.$$

The optimal bound is then

$$\begin{aligned} \mathcal{B}(\delta, \delta', \beta^*) &= B^{\text{out}}\delta' + \sqrt{B^{\text{in}}(\delta + \delta')} \left( \sqrt{C} - \sqrt{B^{\text{in}}(\delta + \delta')} \right) + \sqrt{CB^{\text{in}}(\delta + \delta')} \\ &\leq B^{\text{out}}\delta' + 2\sqrt{CB^{\text{in}}(\delta + \delta')}. \end{aligned}$$

$\square$

<sup>2</sup>More precisely the one before the last upper bound in the proof.

## D Forward finite difference estimators

So far, we have considered the following finite difference estimator:

$$\widehat{\nabla}_\theta = \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_\beta, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_0, \theta, 0) \right).$$

It uses two points to estimate the derivative  $d_\beta \partial_\theta \mathcal{L}(\phi_{\theta, \beta}^*, \theta, \beta)$  at  $\beta = 0$ . However, as highlighted by our theoretical analysis, the finite difference error generated by this estimator quickly increases with  $\beta$ . It is possible to reduce this error by adding more points to the estimator, either by evaluating the function before, after or on both sides of the  $\beta$  value we want to estimate the derivative at (here 0).

The latter, called the symmetric version, was introduced in the Hopfield energy setting and was crucial for scaling equilibrium propagation to deeper networks [64]. More precisely, it estimates  $\nabla_\theta$  with

$$\widehat{\nabla}_\theta^{\text{sym}} = \frac{1}{2\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_\beta, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_{-\beta}, \theta, -\beta) \right)$$

where  $\hat{\phi}_\beta$  and  $\hat{\phi}_{-\beta}$  are approximate minimizers of  $\phi \mapsto \mathcal{L}(\phi, \theta, \beta)$  and  $\phi \mapsto \mathcal{L}(\phi, \theta, -\beta)$ , obtained by starting the minimization process at  $\hat{\phi}_0$ , an estimate of  $\phi_{\theta, 0}^*$ . This symmetric finite difference estimator can be extended to include more points and get a finer approximation.

We argue that, given the form of  $L^{\text{in}}$  and  $L^{\text{out}}$ , it is more stable to only take positive  $\beta$  values, and thus use forward finite difference estimators. The reason is the following. As we want to minimize both  $L^{\text{in}}$  and  $L^{\text{out}}$  we can expect them to be bounded from below, hence  $\mathcal{L}$  also is when  $\beta$  is positive. However, this guarantee no longer holds when taking negative  $\beta$  values.

We introduce the  $p$ -forward finite difference estimator, defined as follows:

$$\widehat{\nabla}_\theta^p = \frac{1}{\beta} \sum_{i=0}^{p-1} \alpha_i \frac{\partial \mathcal{L}}{\partial \theta}(\hat{\phi}_{i\beta}, \theta, i\beta)$$

where  $\hat{\phi}_{i\beta}$  are approximations of  $\phi_{i\beta}^*$  for  $i \in \{0, \beta, \dots, (p-1)\beta\}$  and  $\alpha$  are coefficients that are uniquely determined [65]. Their first values are:

$p$	$\alpha_0$	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$
2	-1	1	0	0	0
3	-3/2	2	-1/2	0	0
4	-11/6	3	-3/2	1/3	0
5	-25/12	4	-3	4/3	-1/4

Note that the two points estimator we have used so far is in fact the  $p$ -forward finite difference estimator for  $p = 2$ .

We now sketch an analysis of the meta-gradient estimation error made by those more general estimators. In our analysis of the  $p = 2$  case, we introduced two quantities: the fixed point approximation induced error and the finite difference error. The estimators that we have here presented are designed to decrease the finite difference error. When  $\beta$  is close to 0, we can show that the finite difference error is  $O(\beta^{p-1})$  using Taylor's theorem. For high  $\beta$  values, it is upper bounded as the  $\alpha$  coefficients verify the property

$$\sum_{i=0}^{p-1} \alpha_i = 1.$$

Inspired by the  $\beta/(1+\beta)$  bound we have proved for the  $p = 2$  case in Theorem 2, we can reasonably expect the finite difference error to be upper bounded by a constant times  $\beta^{p-1}/(1+\beta)^{p-1}$ .

Increasing the number of points thus improves the finite difference error. However, it has a conflicting effect: it may increase the fixed point approximation induced error. If Assumption 1. $i$  holds and if the fixed points  $\phi_{i\beta}^*$  are approximated up to precision  $\delta_i$ , this error is upper bounded by

$$\frac{B^{\text{in}}}{\beta} \sum_{i=0}^{p-1} |\alpha_i| \delta_i + B^{\text{out}} \sum_{i=0}^{p-1} i |\alpha_i| \delta_i.$$

The quantities  $\sum |\alpha_i|$  and  $\sum i|\alpha_i|$  quickly (more than linearly) increase with  $p$ , so is the error associated to the approximation of the fixed points. Because of those two conflicting phenomena, the benefits of those estimators have to be evaluated in practice. We do so in Appendix E on a toy quadratic model. We leave the investigation of the  $p$ -forward estimator on more complex models for future work.

## E An analytically tractable model

**Problem setting.** We investigate a quadratic model in which everything can be calculated in closed form and where the assumptions needed for the theory hold. Define  $L^{\text{in}}$  and  $L^{\text{out}}$  as follows<sup>3</sup>:

$$\begin{aligned} L^{\text{in}}(\phi, \theta) &= \frac{1}{2}(\phi - \phi^t)^\top H(\phi - \phi^t) + \frac{\lambda}{2}\|\phi - \theta\|^2 \\ L^{\text{out}}(\phi) &= \frac{1}{2}(\phi - \phi^v)^\top H(\phi - \phi^v) \end{aligned}$$

where  $\lambda$  is a scalar that controls the strength of the regularization,  $\phi^t$  and  $\phi^v$  two vectors and  $H$  a positive definite diagonal matrix. This model is a quadratic approximation of the few-shot learning setting for implicit meta-learning algorithms introduced in the iMAML paper [26] and is an instance of our synaptic model. The rationale behind this approximation is the following: the training and the validation loss share the same curvature but have different minimizers, respectively  $\phi^t$  and  $\phi^v$ . The matrix  $H$  then models the Hessian and we consider it diagonal for simplicity. Thanks to the quadratic approximation, many quantities involved in our contrastive meta-learning rule can be calculated in closed form.

**Calculation of the finite difference error.** A formula for the minimizer of  $\mathcal{L} = L^{\text{in}} + \beta L^{\text{out}}$  can be derived analytically. The derivative of  $\mathcal{L}$  vanishes if and only if

$$((1 + \beta)H + \lambda \text{Id})\phi - H\phi^t - \beta H\phi^v - \lambda\theta = 0,$$

hence

$$\phi_{\theta, \beta}^* = ((1 + \beta)\text{Id} + \lambda H^{-1})^{-1}(\phi^t + \beta\phi^v + \lambda H^{-1}\theta).$$

$\lambda H^{-1}$  is an interesting quantity in this example. It acts as the effective per-coordinate regularization strength: regularization will be stronger on flat directions.

The meta-gradient calculation follows. As

$$\partial_\theta \mathcal{L}(\phi, \theta, \beta) = -\lambda(\phi - \theta),$$

the use of equilibrium propagation theorem (Theorem 3) yields

$$\nabla_\theta = -\lambda \left. \frac{d\phi_{\theta, \beta}^*}{d\beta} \right|_{\beta=0}.$$

It now remains to calculate the derivative of  $\phi_{\theta, \beta}^*$  with respect to  $\beta$  using the formula of  $\phi_{\theta, \beta}^*$ :

$$\begin{aligned} \frac{d\phi_{\theta, \beta}^*}{d\beta} &= ((1 + \beta)\text{Id} + \lambda H^{-1})^{-1} \phi^v \\ &\quad - ((1 + \beta)\text{Id} + \lambda H^{-1})^{-2} (\phi^t + \beta\phi^v + \lambda H^{-1}\theta) \\ &= ((1 + \beta)\text{Id} + \lambda H^{-1})^{-2} ((1 + \beta)\phi^v + \lambda H^{-1}\phi^v - \phi^t - \beta\phi^v - \lambda H^{-1}\theta) \\ &= ((1 + \beta)\text{Id} + \lambda H^{-1})^{-2} ((\phi^v - \phi^t) + \lambda H^{-1}(\phi^v - \theta)) \end{aligned}$$

Define  $\psi := (\phi^v - \phi^t) + \lambda H^{-1}(\phi^v - \theta)$ ; the meta-gradient finally is

$$\nabla_\theta = -\lambda(\text{Id} + \lambda H^{-1})^{-2}\psi.$$

---

<sup>3</sup>In our experiments, we take the dimension of the parameter space  $N$  to be equal to 50. The Hessian is taken to be  $\text{diag}(1, \dots, 1/N)$ .  $\theta$  is randomly generated according to  $\theta \sim \mathcal{N}(0, \sigma_\theta)$  with  $\sigma_\theta = 2$ .  $\phi^t$  and  $\phi^v$  are drawn around  $\phi^\tau \sim \mathcal{N}(0, \sigma_\tau)$  (with  $\sigma_\tau = 1$ ) with a standard deviation of 0.2.



We can now calculate the finite difference error. Recall the equilibrium propagation estimate at fixed points

$$\widehat{\nabla}_{\theta}^* = \frac{1}{\beta} \left( \frac{\partial \mathcal{L}}{\partial \theta}(\phi_{\theta,\beta}^*, \theta, \beta) - \frac{\partial \mathcal{L}}{\partial \theta}(\phi_{\theta,0}^*, \theta, 0) \right).$$

In this formulation, it is equal to

$$\begin{aligned} \widehat{\nabla}_{\theta}^* &= -\frac{\lambda}{\beta}(\phi_{\theta,\beta}^* - \phi_{\theta,0}^*) \\ &= -\lambda \left( (\text{Id} + \lambda H^{-1})((1 + \beta)\text{Id} + \lambda H^{-1}) \right)^{-1} \psi \\ &= (\text{Id} + \lambda H^{-1}) \left( (1 + \beta)\text{Id} + \lambda H^{-1} \right)^{-1} \nabla_{\theta}. \end{aligned}$$

The finite difference can now be lower and upper bounded. First,

$$\nabla_{\theta} - \widehat{\nabla}_{\theta}^* = \beta \left( (1 + \beta)\text{Id} + \lambda H^{-1} \right)^{-1} \nabla_{\theta}.$$

Introduce  $\mu$  the smallest eigenvalue of  $H$  and  $L$  its largest one. We then have

$$\frac{\mu\beta}{(1 + \beta)\mu + \lambda} \|\nabla_{\theta}\| \leq \|\nabla_{\theta} - \widehat{\nabla}_{\theta}^*\| \leq \frac{L\beta}{(1 + \beta)L + \lambda} \|\nabla_{\theta}\|. \quad (8)$$

This shows that the finite difference error part of Theorem 2 is tight and, in this case, accurately describes the behavior of the finite difference error.

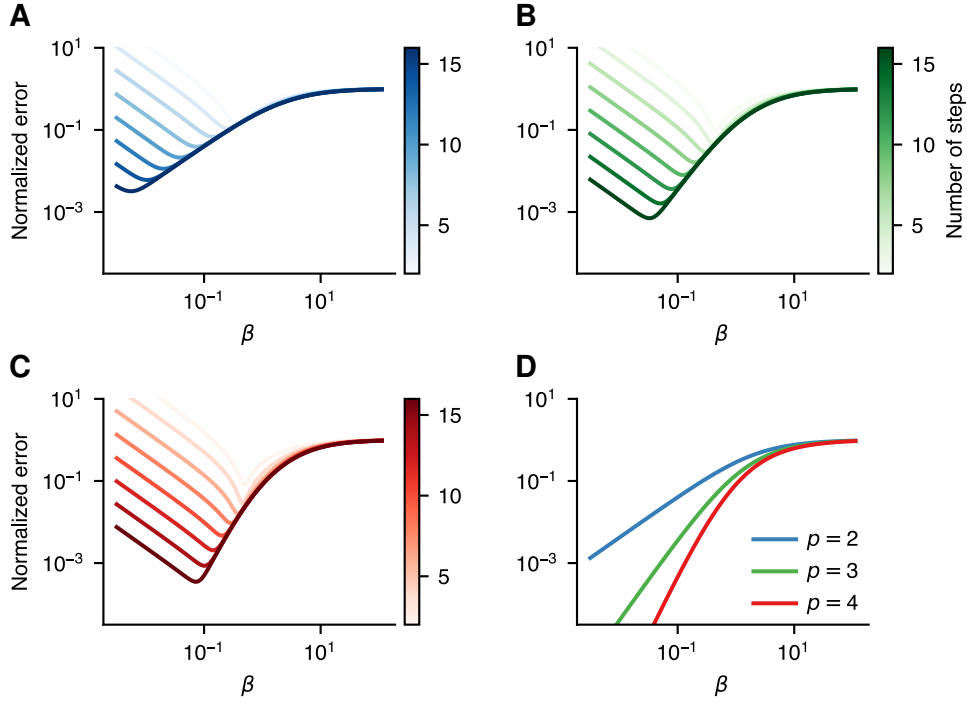


Figure S3: (A, B, C) Normalized error between equilibrium propagation  $p$ -forward finite difference estimators and the outer gradient, as a function of  $\beta$ . The values of  $p$  are (A)  $p = 2$ , (B)  $p = 3$  and (C)  $p = 4$ . (D) Comparison of the finite difference error, as a function of  $\beta$ , for the same three estimators.

**Empirical results.** The fixed point approximation induced error part of the bound cannot be treated analytically as it depends on  $\delta$  and  $\delta'$ , which are, by essence, two empirical quantities. We cannot directly control them either. Instead we use the number of gradient descent steps as a proxy, that it is closely related to  $-\log \delta$  when gradient descent has a linear convergence rate. We choose the number of steps to be the same in the two phases, for the sake of simplicity, even though it may not be optimal. We plot the evolution of the normalized error

$$\frac{\|\nabla_{\theta} - \widehat{\nabla}_{\theta}\|}{\|\nabla_{\theta}\|}$$

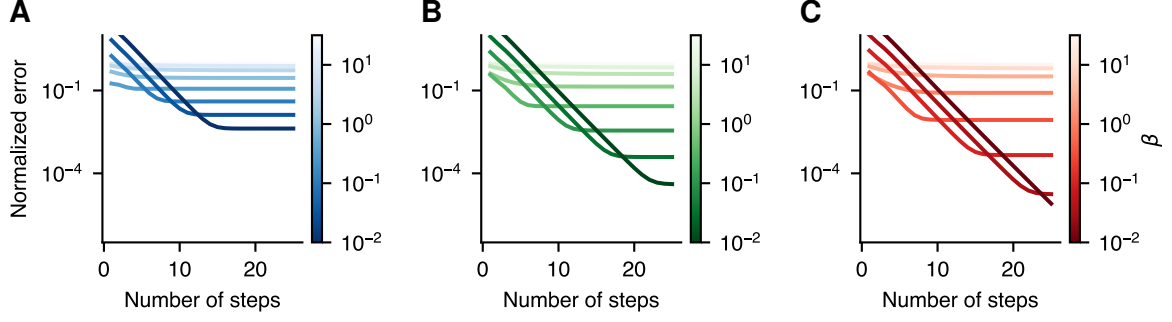


Figure S4: Normalized error between the equilibrium propagation  $p$ -forward finite difference estimate as a function of the number of steps in each phase, for (A)  $p = 2$ , (B)  $p = 3$  and (C)  $p = 4$ .

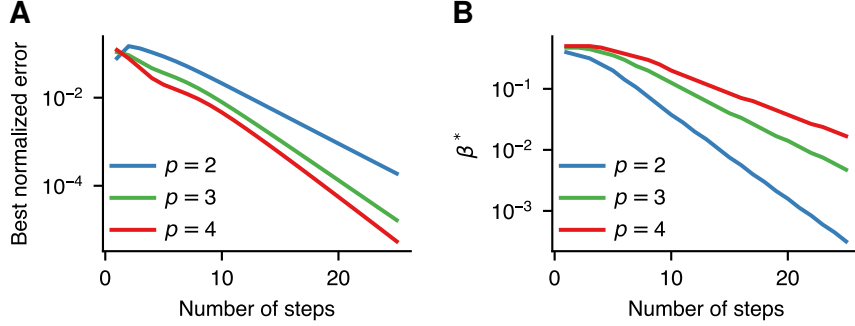


Figure S5: (A) Best normalized error (over the choice of  $\beta$ ) for different  $p$ -forward finite difference estimators, as a function of the number of steps in the two phases. (B)  $\beta^*$  value that is used to obtain the best error.

between the meta-gradient and its equilibrium propagation estimate as a function of  $\beta$  in Fig. S3 for different forward finite difference estimates. The error for  $p = 2$  behaves similarly to what is predicted by the theory (compare with Fig. 1A). The evolution of the error for the other estimators follows the intuition we have developed in Appendix D. The finite difference error saturates when  $\beta$  is high and it converges to 0 faster as  $p$  increases, following the  $O(\beta^{p-1})$  behavior for small  $\beta$  obtained with a Taylor approximation. Note that the comparison of the different forward finite estimators is done for the same number of gradient descent steps in each phase. The 4 points estimator hence uses two times more gradient descent steps than the 2 point estimator. We did this to maintain a relatively similar  $\delta$  values for all the phases, that models some lower bound on the precision to which we can approximate the fixed point in more complex settings. The first three plots would therefore change if we control the total number of steps instead of the number of steps in each phase. The last one won't: it does not depend on the number of steps anymore.

The evolution of the meta-gradient estimation error with the number of steps fits the theoretical predictions as shown on Fig. S4. The benefits brought by longer phases saturate, for every fixed  $\beta$ . When the number of points in the estimation increases, the estimation gets better, as confirmed on Fig. S5A. Note that, from the theoretical analysis we sketched in Appendix D, it was not clear if adding more points to the estimator, would be beneficial. With this plot, we show that it is the case here. In the small error regime (more than 10 steps in this example), the best error and  $\beta$  decrease exponentially with the number of steps, as hinted by Corollary 1. Interestingly, when the number of steps in each phase is fixed, the estimators with more points need higher  $\beta$  value to reach their best performance. This finding complements the purpose for which we introduced them, i.e. decrease the finite difference error. It can be explained in the following way: the more points the more sensitive the forward finite difference estimator is to the fixed point approximation induced error, as we have seen in Appendix D. It is therefore beneficial to increase  $\beta$ , which is made possible by a lower finite difference error.

We finish the study of this quadratic model by considering how the classical estimator performs when the total number of gradient descent steps is fixed (Fig. S6). We retrieve the very same behavior as

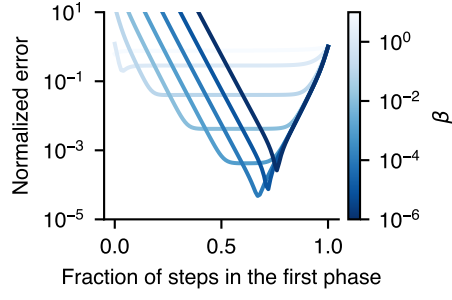


Figure S6: Evolution of the normalized error when the fixed points are computed with gradient descent, as a function of the number of steps in the free phase, for different  $\beta$  values.

for our experiments on the regularization learning setting on the Boston dataset.

## F Hyperparameter search spaces

Tables S1, S2, S3 and S4 show the hyperparameter search spaces for the high-dimensional regularization learning experiments.

Table S1: Hyperparameter search space for CIFAR-10 experiment with 50 training samples and 950 validation samples. For contrastive meta-learning a random search with 500 samples from the grid was performed, for T1T2 the grid was searched exhaustively for 2 independent seeds. Best found parameters are marked in bold.

	Contrastive meta-learning	T1T2
<b>beta</b>	{0.01, 0.03, 0.1, 0.3, 1.0, <b>3.0</b> , 10.0}	-
<b>init-l2</b>	0.001	0.001
<b>lr-inner</b>	0.001	{0.0001, 0.0003, <b>0.001</b> , 0.003, 0.01, 0.03, 0.1}
<b>lr-outer</b>	{0.0001, 0.0003, <b>0.001</b> , 0.003, 0.01, 0.03, 0.1}	{0.0001, 0.0003, <b>0.001</b> , 0.003, 0.01, 0.03, 0.1}
<b>steps-inner</b>	2000	{500, 1000, <b>2000</b> }
<b>steps-outer</b>	100	100
<b>steps-nudged</b>	{ <b>20</b> , 100, 200, 500}	-

Table S2: Hyperparameter search space for MNIST experiment with 10 training samples and 9990 validation samples. For both methods a random search with 500 samples from the grid was performed. Best found parameters are marked in bold.

	Contrastive meta-learning	T1T2
<b>beta</b>	{0.01, 0.03, 0.1, 0.3, 1.0, <b>3.0</b> , 10.0}	-
<b>init-l2</b>	0.001	0.001
<b>lr-inner</b>	{0.0001, 0.0003, 0.001, <b>0.003</b> , 0.01, 0.03, 0.1}	{ <b>0.0001</b> , 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1}
<b>lr-outer</b>	{0.0001, 0.0003, <b>0.001</b> , 0.003, 0.01, 0.03, 0.1}	{0.0001, 0.0003, 0.001, 0.003, 0.01, <b>0.03</b> , 0.1}
<b>steps-inner</b>	2000	{100, 200, 500, 1000, <b>2000</b> }
<b>steps-outer</b>	100	100
<b>steps-nudged</b>	{20, 50, <b>100</b> , 200, 500}	-

Table S3: Hyperparameter search space for MNIST experiment with 1 training sample and 9999 validation samples. For both methods 500 samples were randomly drawn from the search space. Best found parameters are marked in bold.

	Contrastive meta-learning	T1T2
<b>beta</b>	{0.01, 0.03, 0.1, 0.3, 1.0, <b>3.0</b> , 10.0}	-
<b>init-l2</b>	0.001	0.001
<b>lr-inner</b>	Log Uniform(0.0001, 0.1) $\rightarrow$ <b>0.00865</b>	Log Uniform(0.0001, 0.1) $\rightarrow$ <b>0.00027</b>
<b>lr-outer</b>	Log Uniform(0.0001, 0.1) $\rightarrow$ <b>0.0031</b>	Log Uniform(0.0001, 0.1) $\rightarrow$ <b>0.049693</b>
<b>steps-inner</b>	{100, 200, 500, <b>1000</b> , 2000}	{100, <b>200</b> , 500, 1000, 2000}
<b>steps-outer</b>	100	100
<b>steps-nudged</b>	{20, <b>50</b> , 100, 200, 500}	-

Table S4: Hyperparameter search space for full MNIST experiment with 30000 training samples and 30000 validation samples. For both methods, every grid point was evaluated. Best found parameters are marked in bold.

	Contrastive meta-learning	T1T2
<b>beta</b>	{0.01, 0.03, 0.1, 0.3, <b>1.0</b> , 3.0, 10.0}	-
<b>init-l2</b>	0.001	0.001
<b>lr-inner</b>	0.01	{0.0001, 0.0003, 0.001, 0.003, <b>0.01</b> , 0.03, 0.1}
<b>lr-outer</b>	{0.0001, 0.0003, <b>0.001</b> , 0.003, 0.01, 0.03, 0.1}	{ <b>0.0001</b> , 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1}
<b>steps-inner</b>	{1000, <b>2000</b> }	{500, 1000, <b>2000</b> }
<b>steps-outer</b>	100	100
<b>steps-nudged</b>	{20, 50, 100, <b>200</b> , 500}	-