The Lie-Group Bayesian Learning Rule

Eren Mehmet Kıral
RIKEN Center for AI Project

Thomas Möllenhoff
RIKEN Center for AI Project

Mohammad Emtiyaz Khan
RIKEN Center for AI projet

Abstract

The Bayesian Learning Rule provides a framework for generic algorithm design but can be difficult to use for three reasons. First, it requires a specific parameterization of exponential family. Second, it uses gradients which can be difficult to compute. Third, its update may not always stay on the manifold. We address these difficulties by proposing an extension based on Lie-groups where posteriors are parametrized through transformations of an arbitrary base distribution and updated via the group's exponential map. This simplifies all three difficulties for many cases, providing flexible parametrizations through group's action, simple gradient computation through reparameterization, and updates that always stay on the manifold. We use the new learning rule to derive a new algorithm for deep learning with desirable biologically-plausible attributes to learn sparse features. Our work opens a new frontier for the design of new algorithms by exploiting Lie-group structures.

1 INTRODUCTION

The recently proposed Bayesian Learning Rule (BLR) of Khan and Rue (2021) provides a general framework to derive many well-known algorithms from fields such as optimization, deep learning, and graphical models. The rule uses natural-gradient descent to find approximations of the generalized posterior distribution and can recover both Bayesian and non-Bayesian algorithms by employing various exponential-family (EF) distributions. It has been used to design new algorithms, for instance, for uncertainty estimation in deep learning (Khan et al., 2018; Osawa et al., 2019; Lin et al., 2019a; Meng et al., 2020; Möllenhoff and Khan, 2023). Any improvements to the BLR framework can potentially be useful for such algorithm design as well.

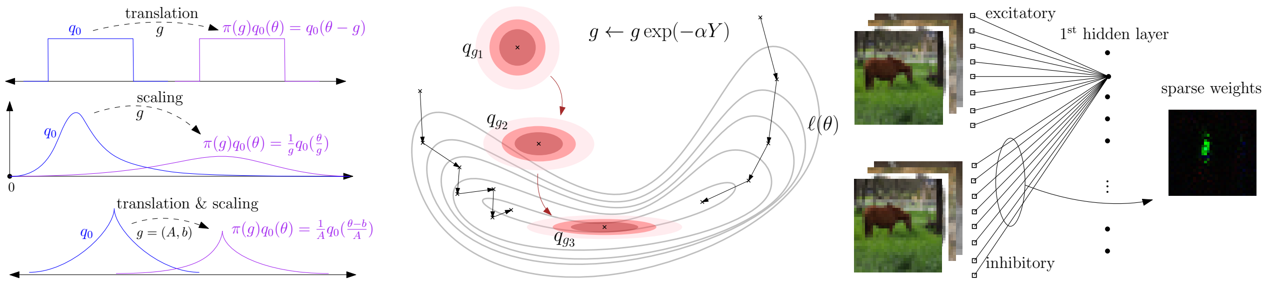
Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

Despite its usefulness, the BLR can be difficult to use for three reasons. First, it relies heavily on pairings of natural and expectation parameters of the EFs which do not naturally exist for generic distributions and can make it difficult to apply the BLR to such cases (Lin et al., 2019a). Second, the BLR requires natural-gradients whose computation is not always straightforward and requires tricks that need to be invented for each specific case, for example, Lin et al. (2019b) use Stein's identity for Gaussians and Meng et al. (2020) use Gumbel-softmax trick for Bernoulli distributions. A last difficulty is that the BLR updates are not always guaranteed to stay within the manifold of distributions, which may require additional modifications (Lin et al., 2020). Our goal here is to address these three difficulties with the BLR.

We propose an extension of the BLR based on Lie-groups where posterior candidates are parametrized through transformations of an arbitrary base distribution by using the group's *action* on the model parameters. For example, the additive group, denoted by $(\mathbb{R}, +)$, translates real scalar parameters by addition, and the multiplicative group, denoted by $(\mathbb{R}_{>0}, \times)$, scales positive scalar parameters by multiplication (Fig. 1a). Many popular distributions can be parameterized this way, including both EF and non-EF distributions (Barndorff-Nielsen et al., 1982, 2012).

We derive a new learning rule called the Lie-group BLR that uses the group's *exponential map* to update candidate distributions (Fig. 1b). A linear approximation of the map coincides with the BLR for some distributions, but the new rule is much easier to use in many cases. First, it does not depend on the EF parameterization but on the Lie-group action which is relatively easier to work with, for example, when using non-EF distributions such as the Laplace distribution. Second, gradient computations are simplified by a simple change of variables to push the derivative to the loss function, giving rise to a general yet easy-to-implement reparameterization trick. Third, due to the closure property of the group, the update naturally stays within the manifold; no additional effort or approximations are required. The new rule also simplifies the computation of the Fisher matrix and inclusion of momentum. Overall, the new learning rule is much easier to use than the BLR.

We show three use-cases for algorithm design in deep



(a) (b) (c)

Figure 1: (a) Posterior candidates are parametrized by transforming the base distribution q_0 over parameter θ. The three figures show the additive, multiplicative, and affine groups, respectively. In each figure, a group element g is applied to q_0 through the action $\pi(g)$, giving rise to a new candidate. The base distribution is set to uniform (top), Rayleigh (middle), and Laplace (bottom) respectively. (b) The Lie-group BLR uses the group's exponential map to update g giving rise to new candidates $q_g = \pi(g)q_0$ (shown with red ovals); an exact update is given in Eq. 10. The algorithm is different from standard gradient methods used in deep learning to learn θ (black arrows). (c) The Lie-group BLR with multiplicative group gives rise to a new algorithm to train neural networks with constraints similar to those found in biological neural networks. Specifically, the signs of the weights from a node are fixed, making the node either excitatory or inhibitory. This gives rise to sparse and localized features similar to those found in the receptive fields of the mammalian visual cortex.

learning by employing the additive, multiplicative, and affine groups respectively. The additive and affine groups result in algorithms similar to those used in deep learning, but the multiplicative group gives rise to a new kind of algorithm to train neural networks with biologically-plausible attributes. We consider networks with nodes that are forced to be either excitatory or inhibitory by fixing the signs of their weights (Fig. 1c). This aims to mimic constraints such as those observed in the receptive fields of mammalian visual cortex (Hubel and Wiesel, 1962; Olshausen and Field, 1996). By design, the new algorithm preserves the signs of the weights by keeping each update within the manifold and ends up learning sparse and localized features (Fig. 1c). The use case shows the usefulness of the new learning rule in designing algorithms that encourage explainability, compositionality, and disentanglement (Bernstein et al., 2020; Whittington et al., 2022).

2 THE BAYESIAN LEARNING RULE

Given a loss function $\ell(\theta)$ over a model with parameter $\theta \in \Theta$, the BLR aims to find

$$q_* \in \underset{q \in \mathcal{Q}}{\arg\min}\ \mathbb{E}_q[\ell] - \tau\mathcal{H}(q), \tag{1}$$

where q is a posterior candidate, $\mathcal{Q}$ is a space of candidate distributions, $\mathcal{H}(q) = -\int_\Theta q(\theta)\log q(\theta)\,\mathrm{d}\theta$ is the differential Shannon entropy, and $\tau > 0$ is a scalar parameter, sometimes referred to as the temperature. The first term favors regions with low losses, while the second term favors higher spread of q, and balancing them requires an exploration-exploitation tradeoff, favoring flatter regions of low loss. The problem can also rewritten as an inference problem where we seek the best possible posterior candidate in $\mathcal{Q}$ by minimizing the Kulluback-Leibler divergence,

$$\mathcal{E}(q) = \mathbb{D}(q\|p_\tau)$$

where $p_\tau(\theta) \propto e^{-\frac{1}{\tau}\ell(\theta)}$ is the Gibbs posterior, sometimes referred to as the generalized posterior (Catoni, 2007). When the loss corresponds to the log-joint distribution of a Bayesian model, $\mathcal{Q}$ is set to the space of all distribution and $\tau = 1$, the solution in (1) coincides with the posterior distribution; see Zellner (1988). Another interpretation is as a stochastic relaxation where the temperature τ is used to search for suitable minima (Geman and Geman, 1984). Such principles are commonly used in random search (Baba, 1981), stochastic optimization (Spall, 2005), evolutionary strategies (Beyer, 2001), global-optimization methods (Leordeanu and Hebert, 2008), and reinforcement learning (Williams and Peng, 1991; Mnih et al., 2016).

The BLR is a natural-gradient descent (NGD) algorithm to solve (1), and Khan and Rue (2021) show that it can recover well-known algorithms from a variety of fields. Specifically, they use *minimal* exponential-family (EF) distributions of the form $q(\theta) \propto \gamma(\theta)e^{\langle\lambda,\phi(\theta)\rangle}$ parameterized by its natural parameter λ, where $\gamma(\theta)$ is a base measure, $\phi(\theta)$ is a sufficient statistics, and $\langle\cdot,\cdot\rangle$ is an inner product. The BLR solves (1) by updating λ as follows,

$$\lambda \leftarrow \lambda - \alpha\frac{\partial}{\partial\mu}\left[\mathbb{E}_q[\ell] - \tau\mathcal{H}(q)\right] \tag{2}$$

where $\alpha > 0$ is the learning rate and the gradients are taken with respect to the expectation parameter $\mu = \mathbb{E}_q[\phi]$. The BLR can recover many existing algorithms as special cases by simply changing the EF form and employing additional approximations to the gradient. Khan and Rue (2021) show this by deriving gradient descent, Newton's method, and several deep-learning optimizers such as RMSprop and Adam, as well as message passing algorithms, such as, Kalman filters. Design of new algorithms is also possible, for example, for Bayesian deep learning Khan et al. (2018); Osawa et al. (2019); Lin et al. (2019a); Meng et al. (2020); Möllenhoff and Khan (2023).

Despite its usefulness, the BLR can be difficult to use in many cases. First, the BLR update is makes use of the pair (λ, μ), which makes its application difficult for other distributions where such pair is not available. For example, for mixture of EFs, such pairs do not naturally exist, and special restrictions on the distribution are required to derive BLR-like updates; see Lin et al. (2019b). Some headway has been made for curved EFs too, for example, Lin et al. (2021) propose a local parametrization of structured Gaussian covariances, but deriving BLR-style updates for generic distributions remains an open problem.

Second, the gradient with respect to μ is not always straightforward to compute. For Gaussians, we can do this easily by using Stein's identity (Lin et al., 2019b) which reduces the computation to that of $\mathbb{E}_q[\nabla \ell(\theta)]$ and $\mathbb{E}_q[\nabla^2 \ell(\theta)]$ (Khan and Rue, 2021, Eqs. 10-11). However, this trick does not generalize to arbitrary distributions. One option is to compute separately the Fisher and gradient with respect to λ (Khan and Lin, 2017, App. F) but this does not work well in practice due to large size of the Fisher matrix and also numerical difficulties arising due to noisy Fisher when estimated using minibatches; see Salimans and Knowles (2013). Third, λ obtained by (2) may not always be valid natural parameters, that is, the steps might go outside the EF manifold. Khan et al. (2018, App. D1) discuss this problems for Gaussians where the update may result in negative variances. The problem is solved in Lin et al. (2020) by using Riemannian gradient descent but such solutions need to be custom designed for specific cases which is tedious and cumbersome.

3 THE LIE-GROUP BAYESIAN LEARNING RULE

In this paper, we address the difficulties of the BLR described in the previous section by proposing a Lie-group based extension of the BLR. We start by describing Lie groups and their actions, followed by parameterization and exponential map, and finish the section by deriving the new learning rule. Readers unfamiliar with Lie groups can refer to Lee (2013, Chapter 7, 8, and 20) for a detailed study.

3.1 Lie groups and their actions

We denote by $(G, *)$ a Lie-group, where G is a set with a binary operation $*$ satisfying the properties of associativity, existence of an identity element and inverses. These mean three things: first, $g * (h * k) = (g * h) * k$ for all $g, h, k \in G$; second, there exists an identity element $e \in G$ such that $e * g = g * e = g$ for all $g \in G$; finally, for any $g \in G$ there exists an inverse element which we denote by $g^{-1} \in G$ such that $g * g^{-1} = g^{-1} * g = e$. The Lie-group is a group, a smooth manifold, and both of its binary group-operation and inversion are smooth. For groups written with a multiplicative notation, it is common to write gh in place of $g * h$. A smooth manifold is locally diffeomorphic to Euclidean space, that is, there are infinitely differentiable invertible mappings between local patches of G and $\mathbb{R}^m$, where m is called the dimension of the manifold.

As an example, consider $(\mathbb{R}, +)$ where $\mathbb{R}$ is a 1-dimensional smooth manifold and, together with addition, it makes a Lie group. The identity element is 0 and inverse of a given element $x \in \mathbb{R}$ is written as $-x$. Another example is the set of positive reals with multiplication, which forms the group $(\mathbb{R}_{>0}, \times)$. A useful property is that if G_1 and G_2 are two Lie groups, then their Cartesian product $G_1 \times G_2$ is also a Lie group. The definition is extended to vector by repeating it P times to get Lie-groups $(\mathbb{R}^P, +)$ and $(\mathbb{R}^P_{>0}, \times)$, where the addition and multiplication are both applied component-wise.

Given a manifold Θ of parameters θ, we can define the *action* of the Lie group on Θ. The action is a smooth map $G \times \Theta \to \Theta$ mapping every $g \in G$ and $\theta \in \Theta$ as $(g, \theta) \mapsto g \cdot \theta$, where '$\cdot$' denotes an operation satisfying $(gh) \cdot \theta = g \cdot (h \cdot \theta)$ and $e \cdot \theta = \theta$. As an example, consider the group $(\mathbb{R}^P, +)$ and parameter-space $\theta \in \mathbb{R}^P$, then the action is the map $g \cdot \theta = g + \theta$. Here, both G and Θ are the same space $\mathbb{R}^P$, but they can also be different. For example, consider the affine group where $G = \mathrm{Aff}_P(\mathbb{R})$, consisting of pairs (A, b) with A an invertible matrix of size $P \times P$ and $b \in \mathbb{R}^P$, and the group operation given by $(A_1, b_1)(A_2, b_2) = (A_1 A_2, A_1 b_2 + b_1)$. Say $\Theta = \mathbb{R}^P$, same as before. The action of G on Θ is then given as $(A, b) \cdot \theta = A\theta + b$, for any $g = (A, b) \in G$ and $\theta \in \Theta$.

3.2 Lie group parametrization

Using the action of G on Θ, we can define another action on the space of measures by pushforwards. To be precise, given a measure ν on Θ and a measurable set $A \subseteq \Theta$ we define $\pi(g)\nu(A) = \nu(g^{-1}A)$ where $gA = \{g \cdot \theta : \theta \in A\}$. Considering probability measures of the form $\nu = q(\theta)\, \mathrm{d}\theta$, in terms of the probability density functions, we have

$$(\pi(g)q)(\theta) = \left| \frac{\mathrm{d}(g \cdot \theta)}{\mathrm{d}\theta} \right|^{-1} q(g^{-1} \cdot \theta), \tag{3}$$

where $\mathrm{d}(g \cdot \theta)/\, \mathrm{d}\theta$ is the Jacobian determinant of $\theta \mapsto g \cdot \theta$.

We take a base distribution given with positive density q_0, and let the space of candidate distribution $\mathcal{Q}$ be the orbit of q_0 under the action of G, defined below,

$$\mathcal{Q} = \{\pi(g)(q_0(\theta)\, \mathrm{d}\theta) : g \in G\}. \tag{4}$$

This gives us a transitive action of G on $\mathcal{Q}$. Also every $q \in \mathcal{Q}$ can be parametrized by group elements g to write $q = q_g = \pi(g)q_0$. We will denote this parametrization by $\varphi : G \to \mathcal{Q}$ with $\varphi(g) = q_g$.

Here is an example. Take $q_0(\theta) = \theta e^{-\frac{1}{2}\theta^2}$, which is a parameter-free distribution on $\Theta = \mathbb{R}_{>0}$. The group $(\mathbb{R}_{>0}, \times)$ acts on Θ by $g \cdot \theta = g\theta$, and the Jacobian of this map is simply g. The pushforward action of $g \in \mathbb{R}_{>0}$ on q_0 traverses the set of Rayleigh distributions

$$\mathcal{Q} = \left\{ q_g(\theta) = \tfrac{\theta}{g^2} e^{-\frac{\theta^2}{2g^2}} : g \in \mathbb{R}_{>0} \right\}, \tag{5}$$

which is a family parametrized by the group.

The parameterization depends on the action over a group element g and is different from those used for EF. The good news is that many EFs can be parameterized this way, for example, Gaussian and Bernoulli distribution. These are also sometimes referred to as the transformation families or models (Barndorff-Nielsen et al., 2012). The advantage of this parameterization is that it can be relatively easier to work with when using non-EF distributions such as the Laplace distribution. This will be useful to extend the BLR.

3.3 The exponential map and Lie group updates

Given the group parametrization above, our goal is to find a group element g_* such that

$$g_* \in \arg\min_{g \in G} \mathcal{E}(q_g).$$

We can find g_* with an iterative update, for example, by slowly moving in the direction of fastest descent that leads us to g_*. This can be done by using the exponential map.

Given a Lie group, its tangent space at identity, denoted by T_eG, is called the Lie algebra of G; see the first figure in Fig. 2. The exponential map is a smooth function 'folding' the tangent space at identity to the group, which we denote by $\exp : T_eG \to G$. The map is well defined for all tangent vectors, and in fact it is one-to-one and onto in small neighborhoods around the $\mathbf{0} \in T_eG$ vector and $e \in G$. As an example, for matrix groups the exponential map is given by the Taylor series $\exp(X) = \sum_{n=0}^{\infty} \frac{X^n}{n!}$. For diagonal matrices $X = \operatorname{diag}(\lambda_1, \dots, \lambda_r)$ we can easily calculate $\exp(X) = \operatorname{diag}(e^{\lambda_1}, \dots, e^{\lambda_r})$.

For any $X \in T_eG$ the exponential map defines paths $\gamma_X : \mathbb{R} \to G$ in G via $\gamma_X(t) := \exp(tX)$ satisfying $\gamma_X(t_1)\gamma_X(t_2) = \gamma_X(t_1 + t_2)$. It is a path going through the identity at $t = 0$ in the direction of X, meaning $\gamma_X(0) = e$ and $\frac{\mathrm{d}}{\mathrm{d}t}\gamma_X(t)\big|_{t=0} = X$. At a particular $g \in G$, we can use an update of the form

$$g \leftarrow g \exp(-\alpha X),$$

moving in the direction of $X \in T_eG$ by a step-size of $\alpha > 0$. The direction can be the one with the fastest ascent.

3.4 Simplifying gradients through reparametrization

How to find the direction of fastest ascent? A summary is given in Fig. 2. Essentially, we have to connect $X \in T_eG$ to the vectors in $\mathcal{Q}$. We will see that the computation will be simplified by using a change of variable to push the derivative to the loss function, giving rise to a general yet easy-to-implement reparameterization trick.

Let $T_{q_g}\mathcal{Q}$ denote the tangent space in $\mathcal{Q}$ at a point q_g. We first parametrize $T_{q_g}\mathcal{Q}$ by the vectors in the Lie algebra and then compute the differential of $\mathcal{E}$ on vectors expressed in this way. For every $g \in G$ the left multiplication map $L_g : G \to G$, defined as $L_g(h) = gh$, is an invertible smooth map on the manifold with its inverse being $L_{g^{-1}}$. These global diffeomorphisms give us linear maps $\mathrm{d}L_g(e) : T_eG \to T_gG$ between the vector spaces. The map $\mathrm{d}\varphi(g)$ then sends it to $T_{q_g}\mathcal{Q}$. Therefore call, $h_X^g = \mathrm{d}(\varphi \circ L_g)X \in T_{q_g}\mathcal{Q}$; see Fig. 2 for a visualization of these vectors and mappings. The perturbations of $\mathcal{E}$ at q_g are given by the tangent vectors

$$h_X^g(\theta) = \frac{\mathrm{d}}{\mathrm{d}t} q_{ge^{tX}}(\theta)\bigg|_{t=0} \in T_{q_g}\mathcal{Q}.$$

An explicit computation of these tangent vectors is given in the Appendices A.3 to A.5 for certain Lie groups.

The linear map $X \mapsto h_X^g$ is surjective. Also, notice that the tangent vectors h_X are integrable functions on Θ satisfying $\int_\Theta h_X = 0$, since $\int_\Theta q_{ge^{tX}}(\theta)\,\mathrm{d}\theta$ is constant (equal to 1) for all t. In what follows, we will drop g from the notation h_X^g whenever it is clear that we are working in the tangent space at q_g or whenever g does not matter.

We show in App. A.1 that the integral form of the differential can be written as follows,

$$(\mathrm{d}\mathcal{E}(q_g))[h_X] = \int h_X(\theta) \log \frac{q_g(\theta)}{p_\tau(\theta)}\,\mathrm{d}\theta, \tag{6}$$

where $\mathrm{d}\mathcal{E}(q_g) : T_{q_g}\mathcal{Q} \to \mathbb{R}$ measures the change in $\mathcal{E}$ when q_g is perturbed by variations h_X. The integral has two parts

$$(6) = \underbrace{\frac{1}{\tau}\int h_X(\theta)\ell(\theta)\,\mathrm{d}\theta}_{①} + \underbrace{\int h_X(\theta)\log q_g(\theta)\,\mathrm{d}\theta}_{②}. \tag{7}$$

Only the first part depends on the loss function ℓ and the second part is the differential of the entropy term in (1).

In calculating ① and ② we will come across the infinitesimal action of G, which is a map from T_eG to vector fields on Θ. To be precise, we define $X \cdot \theta := \frac{\mathrm{d}}{\mathrm{d}t}(\exp(tX) \cdot \theta)\big|_{t=0}$. Note that for each $\theta \in \Theta$ the right hand side is a tangent vector in $T_\theta\Theta$. For example if the parameter space is $\Theta = \mathbb{R}^P_{>0}$ then $X \cdot \theta \in T_\theta\Theta \cong \mathbb{R}^P$, i.e. it can have negative entries. It is the variation in the parameters that an infinitesimal variation in the direction X

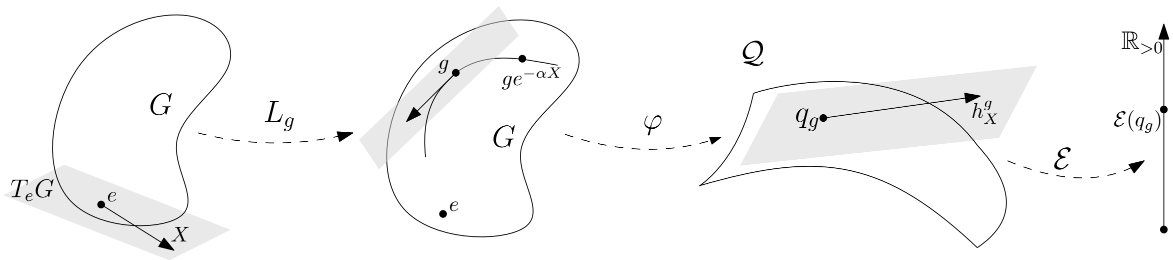


Figure 2: L_g is the multiplication-by-g map and sends the identity element e to g. Its differential takes the vector X at the Lie algebra T_eG to a vector tangent to g. The differential of $\varphi : g \mapsto q_g$ maps it to a tangent vector $h_X^g \in T_{q_g}\mathcal{Q}$. Among all such tangent directions we choose the one corresponding to the direction of fastest ascent for the objective function $\mathcal{E}$. Then we update the g using the exponential map, which defines a curve on G through g in a desired direction $-X$.

will induce. Also for the calculation we need to introduce $\mathrm{Ad}_g : T_eG \to T_eG$ called the adjoint representation of the Lie group, defined as $\mathrm{Ad}_g(X) := \frac{\mathrm{d}}{\mathrm{d}t}(ge^{tX}g^{-1})\big|_{t=0}$. For commutative groups this adjoint representation is clearly the identity, i.e. $\mathrm{Ad}_g(X) = X$ for all $g \in G, X \in T_eG$.

In ①, start with the definition of h_X, differentiate under the integral sign, and calculate

$$\begin{aligned} \text{①} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int q_{ge^{tX}}(\theta)\ell(\theta)\,\mathrm{d}\theta\right)\bigg|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int q_0(\theta)\ell\left(ge^{tX}\cdot\theta\right)\,\mathrm{d}\theta\right)\bigg|_{t=0} \\ &= \int q_0(\theta)(\nabla_\theta \ell(g\cdot\theta))^\top \frac{\mathrm{d}}{\mathrm{d}t}\left(ge^{tX}\cdot\theta\right)\bigg|_{t=0}\mathrm{d}\theta \\ &= \int q_g(\theta)(\nabla_\theta\ell(\theta))^\top\left(\mathrm{Ad}_g(X)\cdot\theta\right)\,\mathrm{d}\theta. \end{aligned} \tag{8}$$

In the second line we applied $\theta \mapsto ge^{tX}\cdot\theta$. After this change of variables note that t is now only in ℓ and a derivative in t will introduce $\nabla_\theta\ell$ via the chain rule. Finally in the last line we do another change of variables $\theta \mapsto g^{-1}\cdot\theta$ by which the infinitesimal action of $\mathrm{Ad}_g(X)$ on θ appears.

The computation is simplified by using the change of variable and the derivative sits on the loss function which can be computed by automatic differentiation techniques. The entropic contribution is calculated similarly, using the same reparametrization technique (App. A.2).

$$\text{②} = \int\left(\nabla_\theta q_0(\theta)\right)^\top(X\cdot\theta)\,\mathrm{d}\theta. \tag{9}$$

Notice that we now have an expectation (over q_g) of a quantity involving $\nabla_\theta\ell$, thus make use of not only the value of the loss ℓ, but also its first order information.

For the translation of these abstract formulas to particular cases, we refer the reader to Sec. 4. Particular bases for $X \in T_eG$ are chosen and these integrals are calculated, giving us our concrete update rules.

3.5 The new learning rule

The Lie-Group BLR uses the following update

$$g \leftarrow g\exp(-\alpha Y) \text{ where } h_Y = \left(\mathrm{d}\mathcal{E}(q_g)\right)^\sharp \in T_{q_g}\mathcal{Q}. \tag{10}$$

Here, $Y \in T_eG$ is such that its image $h_Y^g \in T_{q_g}\mathcal{Q}$ under $\mathrm{d}\varphi\circ\mathrm{d}L_g$ is $(\mathrm{d}\mathcal{E}(g_q))^\sharp$, which is the direction of fastest ascent for the functional $\mathcal{E}$. The update naturally stay within the manifold due to the closure property of the group; the exponential map folds the tangent vector back on to the manifold. We will now explain the terminology used in this update.

Given a differentiable map $F : M \to N$ between two manifolds, its differential is a linear map between the tangent spaces, $\mathrm{d}F(x) : T_xM \longrightarrow T_{F(x)}N$. If $N \subseteq \mathbb{R}$ then $\mathrm{d}F(x)$ is a linear functional taking tangent vectors to real numbers, such an object is called a cotangent vector denoted as $\mathrm{d}F(x) \in T_x^*M$. In vanilla gradient descent on flat space, i.e., if $F : \mathbb{R}^n \to \mathbb{R}$, the differential of the real valued function is a row vector multiplying tangent (column) vectors which spit out numbers, thus it is also a covector. One transposes it to get to get the gradient ∇F, which maximizes the linear approximation $v \mapsto F(x) + (\nabla F)^\top v$ subject to $v^\top v \leq 1$, i.e. it is the direction of fastest ascent with respect to the Euclidean metric. The operator $\sharp$, (called the musical-isomorphism sharp) is the manifold analogue of transposing a row vector to get the gradient, and depends on a choice of metric.

A metric is a positive definite nondegenerate symmetric bilinear form on the tangent spaces of $\mathcal{Q}$. Given such a bilinear form ω, fixing one of the variables we get a linear functional from the tangent space to reals, i.e. a covector. Therefore $\flat : T_q\mathcal{Q} \longrightarrow T_q^*\mathcal{Q}$ such that $(h_Y)^\flat = \omega(\cdot, h_Y)$ is a linear map called *flat*, which is invertible since ω is nondegenerate. The inverse of this isomorphism is the musical-isomorphism called *sharp* denoted by $\lambda^\sharp \in T_q\mathcal{Q}$ for any $\lambda \in T_q^*\mathcal{Q}$.

In vector notation, after choosing a basis $\{h_i\}_{i=1,2,\dots,m}$ for $T_q\mathcal{Q}$ (where $m = \dim(\mathcal{Q})$), the metric is given by

$\omega(v, w) = v^\top F w$ for F the symmetric matrix with entries $F_{ij} = \omega(h_i, h_j)$. Then $\flat$ maps a given (column) vector v to the (row) vector $v^\top F$, and its inverse is $(w^\top)^\sharp = F^{-1} w$.

The differential $\mathrm{d}\mathcal{E}(q_g)$ of the functional $\mathcal{E}$ is a covector in $T^*_{q_g}\mathcal{Q}$. In vector notation, optimizing the linearization $\mathcal{E}(q) + (\mathrm{d}\mathcal{E}(q_g))[h_X]$ of the objective functional $\mathcal{E}$ near q, subject to the condition $\omega(h_X, h_X) \leq 1$ corresponds to

$$\max_{v \in \mathbb{R}^m} d^\top v \quad \text{s.t.} \quad v^\top F v = 1,$$
$$\text{where } d = \begin{bmatrix} (\mathrm{d}\mathcal{E}(q_g))[h_1] & \cdots & (\mathrm{d}\mathcal{E}(q_g))[h_m] \end{bmatrix}^\top . \quad (11)$$

Solving it using Lagrange multipliers we get that v must be a multiple of $F^{-1}d$, which equals $(\mathrm{d}\mathcal{E}(q_g))^\sharp$. Due to (11), this is the direction of fastest ascent with respect to the chosen metric ω.

An advantage of our approach is that it simplifies the computation of the Fisher, which now needs to be computed only once. This is because, with the choice h^g_X as tangent vectors, the metric depends only on X's and is independent of g. Using the Fisher metric is natural because it arises as the second-order differential of our objective function $\mathcal{E}$, which means that the direction of fastest descent is aligned with minimizing the second-order approximation of $\mathcal{E}$. These points are discussed in more detail in App. A.1.

It is also easy to implement momentum. As all of our Y vectors are in $T_e G$, we can simply accumulate previous gradient steps to include momentum as follows,

$$M \leftarrow (1 - \beta) Y + \beta M,$$
$$g \leftarrow g \exp(-\alpha M),$$

where $\beta \in [0, 1)$, and the momentum term is initialized at $M = \mathbf{0} \in T_e G$ and $Y \in T_e G$ is found via (10).

4 NEW ALGORITHMS FOR DEEP LEARNING

We will now show three use-cases of the Lie-group BLR to design new algorithms for deep learning. In the BLR, new algorithms can be designed by changing the form of the EF. For the Lie-group BLR, we can do the same by employing various kinds of Lie-groups. The three examples we will show will use the additive, multiplicative, and affine groups respectively. The additive and affine groups result in algorithms similar to those used in deep learning, but the multiplicative group gives rise to a new kind of algorithm to train neural networks with biologically-plausible attributes.

We will also see that, in some cases, a linear approximation of the map coincides with the BLR; a summary of such results is given in App. A.6. The Lie-group BLR extends the BLR and provides yet another way to design new algorithms by employing various Lie-group.

4.1 The additive group $G = (\mathbb{R}^P, +)$

We start by specifying the Lie-group parameterization. We will assume $\Theta = \mathbb{R}^P$, then G acts on Θ via $g \cdot \theta = \theta + g$. We set $q_0(\theta) = \prod_i \tilde{q}_0(\theta_i) \in \mathcal{P}^+(\Theta)$ where $\tilde{q}_0$ is a density function of an everywhere positive probability distribution on $\mathbb{R}$. The action of G on the parameters induces an action on the probability distributions on Θ which in this case is given as $q_g(\theta) := (\pi(g) \cdot q_0)(\theta) = q_0(\theta - g)$, and

$$\mathcal{Q} := \{q_g(\theta)\, \mathrm{d}\theta : g \in \mathbb{R}^P\} = \{q_0(\theta - g)\, \mathrm{d}\theta : g \in \mathbb{R}^P\}.$$

For the additive group, the Lie-group BLR update of (10) reduces to the following (derivation is in App. A.3),

$$g \leftarrow g - \alpha \mathbb{E}_{q_g}[\nabla_\theta \ell]. \quad (12)$$

This coincides with the update of Khan and Rue (2021, Eq. 7) when $\mathcal{Q}$ is the set of Gaussians with variance 1. Clearly if q_0 is a standard Gaussian, then a translation will generate such Gaussians with the mean parameterized by g.

The Lie-group BLR generalizes the update obtained by Khan and Rue (2021) to an arbitrary base distribution q_0. For this simple case, no linear approximation to the map is necessary to arrive at the BLR. This is because this group's exponential map is trivial, that is, already linear. We can also use distributions such as the uniform distribution, even though it is not an everywhere positive density distribution. Although the derivation does not allow for choosing q_0 as a Dirac delta measure, such a choice will give us the classical gradient descent $\theta \leftarrow \theta - \alpha \nabla \ell(\theta)$.

There is also a connection with anticorrelated noise injection (Orvieto et al., 2022), which has been shown to perform better than gradient descent and its perturbed versions. Assume q_0 is centered around $\mathbf{0} \in \mathbb{R}^P$, so the mean of q_θ is θ. If we use a single MC sample for the expected gradient, then the update rule is

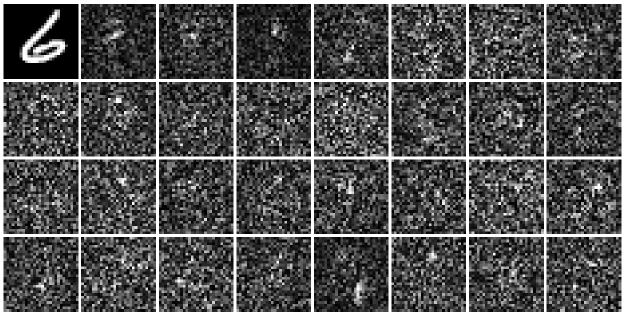
$$\theta \leftarrow \theta - \alpha \nabla \ell(\theta + \xi),$$

with noise $\xi \sim q_0$. This is exactly Orvieto et al. (2022, Eq. 3) when the current iterate θ is set to the mean of q.

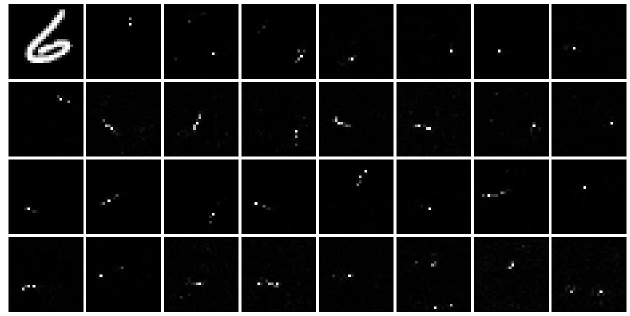
4.2 The multiplicative group $G = (\mathbb{R}^P_{>0}, \times)$

We consider networks with nodes that are forced to be either excitatory or inhibitory by fixing the signs of their weights (Fig. 1c). This aims to mimic constraints such as those observed in the receptive fields of mammalian visual cortex (Hubel and Wiesel, 1962; Olshausen and Field, 1996). We will use the multiplicative group to parameterize the distribution over the weights.

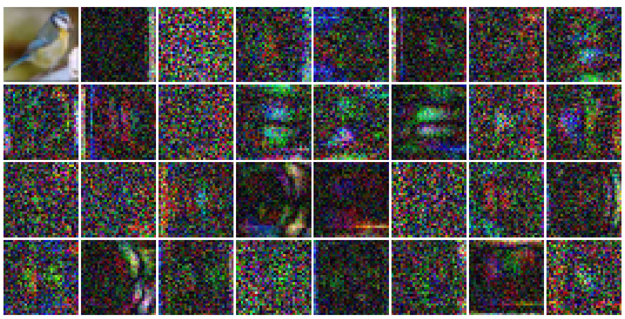
Let the parameter space be $\Theta = \mathbb{R}^P_{>0}$. For example consider the weights of a neural network whose signs are immutable and their magnitude is the only trainable parameter. In fact, we may assign certain nodes as excitatory (respectively inhibitory)—as is the case in biological neural



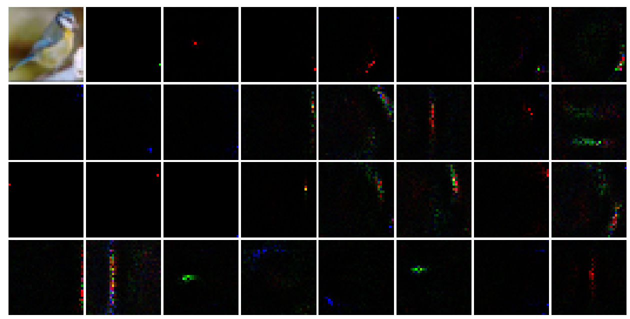
(a) MNIST (MLP), additive, accuracy: 98.38



(b) MNIST (MLP), multiplicative, accuracy: 98.59



(c) CIFAR-10 (MLP), additive, accuracy: 58.85



(d) CIFAR-10 (MLP), multiplicative, accuracy: 59.19

Figure 3: Learning using multiplicative rule (see Alg. 2) on a network with biologically plausible attributes leads to sparse and localized first-layer weights. This is in constrast to the additive rule (Alg. 1). In the figures, we visualize the weights connected to the neurons with highest activation under the input pattern in the top-left frame. The feature-detectors learned by our multiplicative rule show compositional and disentangled traits. This phenomenon occurs both for MNIST (a)–(b) and CIFAR-10 (c)–(d).

circuitry—and set all the signs of weights for connections emanating from a cell as $+$ (respectively $-$). This is also known as the $\pm$ trick, (Ghai et al., 2020). This setup would then respect *Dale's Law* from neurobiology, which is the assumption that a neuron has the same (excitatory or inhibitory) behaviour at all of its synapses, and that this does not change during training or stochastically. In neural networks this corresponds to keeping signs of weights fixed, see Amit et al. (1989) and Beyer et al. (2021).

G acts on Θ by componentwise multiplication and given $q_0 \in \mathcal{P}^+(\Theta)$ as above, the transformations look like $q_g(\theta) := (\pi(g)q_0)(\theta) = \frac{1}{\prod_i g_i} q_0\left(\frac{\theta_1}{g_1}, \ldots, \frac{\theta_P}{g_P}\right)$. The manifold of candidate distributions is $\mathcal{Q} = \{q_g(\theta) : g \in G\}$ and the Lie-group BLR reduces to (derivation in App. A.4)

$$g_i \leftarrow g_i \exp\left(-\alpha\left(\mathbb{E}_{q_g}\left[\theta_i \partial_i \ell\right] - \tau\right)\right). \tag{13}$$

Notice that the parameter conditions $g_i > 0$ are automatically satisfied, thus we stay on the manifold $\mathcal{Q}$. The new algorithm can be used to train networks with desirable biologically-plausible attributes. In our experiments, we find that the algorithm gives sparse features. The new learning rule can be useful to design algorithms that encourage explainability, compositionality, and disentanglement (Bernstein et al., 2020; Whittington et al., 2022).

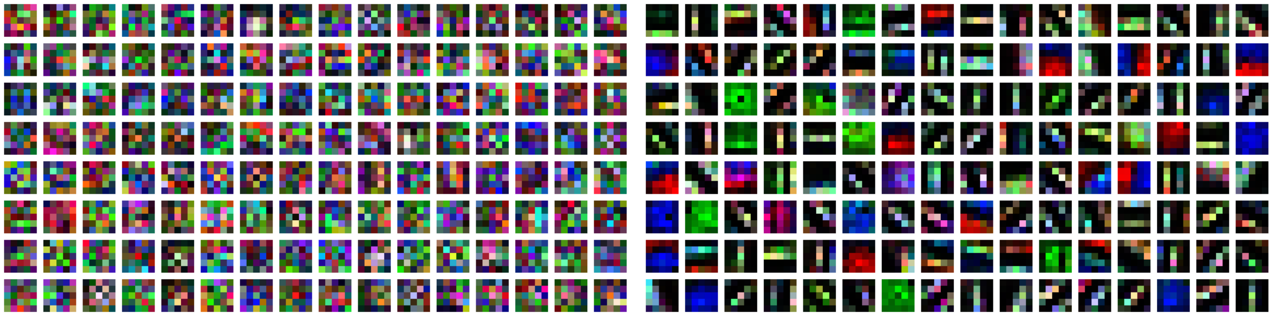
We can show that linearization of (13) recovers the BLR-update for Rayleigh distributions (5). We show this for the 1-dimensional case. This is also an exponential family $\mathcal{Q} = \{q^\lambda(\theta) := \theta\lambda e^{-\frac{1}{2}\theta^2\lambda} : \lambda > 0\}$ where connection to (5) is given as $\lambda = 1/g^2$. For this case, the BLR-update (2) can be written as follows,

$$\begin{aligned}\lambda \leftarrow & \lambda - \alpha {F_\lambda}^{-1} \nabla_\lambda \left[\mathbb{E}_{q^\lambda}[\ell] - \mathcal{H}(q^\lambda)\right] \\ & = \left(1 - \frac{\alpha}{2}\right)\lambda + \frac{\alpha\lambda}{2}\mathbb{E}_{q^\lambda}[\theta\,\partial\ell].\end{aligned} \tag{14}$$

The update (2) uses differentation with respect to the expectation parameters, which we expressed here using the natural parameters and the inverse Fisher as in Khan and Rue (2021, Eq. 4). In passing from first line to the second, we used that the Fisher inverse matrix ${F_\lambda}^{-1}$ is multiplying with λ^2 and we made a change of variables $\theta\sqrt{\lambda} \mapsto \theta$ before differentiating in λ in the second line. On the other hand, square and reciprocate both sides of (13) so that the update is written in terms of λ. This gives (for $\tau = 1$)

$$\begin{aligned}\lambda \leftarrow & \lambda \exp(2\alpha(\mathbb{E}_{q^\lambda}[\theta\,\partial\ell] - 1)) \\ & \approx \lambda(1 + 2\alpha(\mathbb{E}_{q^\lambda}[\theta\,\partial\ell] - 1)),\end{aligned} \tag{15}$$

which coincides with the BLR (14) for a different step-size.



(a) CIFAR-10 (CNN), additive, accuracy: 88.50 (b) CIFAR-10 (CNN), multiplicative, accuracy: 87.55

Figure 4: Similar to the fully-connected setting of Fig. 3, our multiplicative rule leads to sparse and interpretable weights (here: convolutional filters). The weights (filters) learned by the multiplicative rule have a shorter description length. For example, they can often be understood as edge detectors or are monochromatic (e.g., detecting blue or green patterns).

4.3 The diagonal affine group

The action of the affine group combines translations and scaling. This group can be realized as pairs (A, b) for diagonal positive $P \times P$ matrices A and $b \in \mathbb{R}^P$ where the group operation is given by $(A_1, b_1)(A_2, b_2) = (A_1A_2, A_1b_2 + b_1)$ and the group action on Θ is given as $(A, b) \cdot \theta = A\theta + b$.

The exponential map for this group is more complicated. Assuming q_0 is even (a technical assumption made only for a cleaner formula), the update rule is (see App. A.5)

$$b_i \leftarrow b_i + \frac{c_X}{c_y} A_i \frac{\exp(-\alpha U) - 1}{U} V, \tag{16}$$

$$A_i \leftarrow A_i \exp(-\alpha U), \tag{17}$$

where $U = \mathbb{E}_{q_g}[(\theta_i - b_i)\partial_i \ell] - \tau$ and $V = A_i \mathbb{E}_{q_g}[\partial_i \ell]$. Also $c_X = \int_{\mathbb{R}} \left(1 + \theta \frac{\tilde{q}_0'(\theta)}{\tilde{q}_0(\theta)}\right)^2 \tilde{q}_0(\theta)\, \mathrm{d}\theta$, $c_y = \int_{\mathbb{R}} \frac{\tilde{q}_0'(\theta)^2}{\tilde{q}_0(\theta)}\, \mathrm{d}\theta$ are constants that can be calculated once and for all for a given $\tilde{q}_0$. Choosing q_0 as a Dirac delta measure gives us gradient descent as in the additive case.

In this group action if q_0 is chosen as the normal distribution $\mathcal{N}(0, I)$ then $\mathcal{Q}$ is also an exponential family. We show in A.6 that the linear approximation in α to this update is exactly the BLR from Khan and Rue (2021).

5 NUMERICAL EXPERIMENTS

In this section, we compare our Lie-group BLR (10) to existing methods. We always report the performance for the predictive marginal probability $p(y \mid x)$. This can be computed from our optimal group element $g_* \in G$ via the equation $p(y \mid x) = \int p(y \mid g_* \cdot \theta, x)\, q_0(\theta)\, \mathrm{d}\theta$. In practice, we approximate the integral using 32 samples independently drawn from q_0.

Model & Dataset	Method	Accuracy ↑ (higher is better)	NLL ↓ (lower is better)	ECE ↓ (lower is better)
MNIST MLP	add. (Alg. 1)	$98.38_{\pm 0.02}$	$0.083_{\pm 0.001}$	$0.012_{\pm 0.000}$
	mult. (Alg. 2)	$98.59_{\pm 0.02}$	$0.058_{\pm 0.001}$	$0.006_{\pm 0.000}$
CIFAR–10 MLP	add. (Alg. 1)	$58.85_{\pm 0.08}$	$1.236_{\pm 0.002}$	$0.085_{\pm 0.001}$
	mult. (Alg. 2)	$59.19_{\pm 0.07}$	$1.160_{\pm 0.001}$	$0.026_{\pm 0.001}$
CIFAR–10 CNN	add. (Alg. 1)	$88.50_{\pm 0.08}$	$1.091_{\pm 0.007}$	$0.096_{\pm 0.001}$
	mult. (Alg. 2)	$87.55_{\pm 0.06}$	$0.498_{\pm 0.003}$	$0.034_{\pm 0.001}$

Table 1: Additive and multiplicative learning updates give comparable test accuracies. However, they learn very different representations, as shown in Fig. 3. Moreover, learning with the multiplicative rule leads to a much smaller expected calibration error and negative log-likelihoods.

5.1 Additive vs. multiplicative learning

We now compare the properties of the additive and multiplicative group updates from Sec. 4.1 and Sec. 4.2 when applied to neural network training. For a detailed pseudocode of the final algorithm please see Alg. 1 and Alg. 2 in the appendix. We use the additive and multiplicative updates to train a feed-forward neural network with 5 hidden layers (MLP) and a small convolutional net (CNN). The exact architectures and hyperparameters are described in App. B.1. The results are summarized in Table 1.

While both the additive (Gaussian q_0) and multiplicative updates (Rayleigh q_0) lead to comparable test accuracies on the MNIST and CIFAR-10 data sets, the learned neural network weights are drastically different. Multiplicative learning leads to sparse, localized and compositional traits. This highlights how different choices of Lie group can lead to different learning behaviors. The weights of the trained neural networks are visualized in Fig. 3 for the MLP and in Fig. 4 for the CNN. Multiplicative learning also tends to improve the negative-log likelihood (NLL) as well as the expected calibration error (ECE) (Guo et al., 2017).

The sparse nature of the filters in the multiplicative family

Method	Family $\mathcal{Q}$	CIFAR-10			CIFAR-100			TinyImageNet		
		Acc. ↑ (higher is better)	NLL ↓ (lower is better)	ECE ↓ (lower is better)	Acc. ↑ (higher is better)	NLL ↓ (lower is better)	ECE ↓ (lower is better)	Acc. ↑ (higher is better)	NLL ↓ (lower is better)	ECE ↓ (lower is better)
Additive (Alg. 1)	Uniform	$91.07_{\pm 0.08}$	$0.365_{\pm 0.003}$	$0.052_{\pm 0.001}$	$64.29_{\pm 0.09}$	$1.437_{\pm 0.004}$	$0.121_{\pm 0.001}$	$49.34_{\pm 0.14}$	$2.234_{\pm 0.008}$	$0.107_{\pm 0.001}$
	Gaussian	$91.28_{\pm 0.11}$	$0.328_{\pm 0.008}$	$0.045_{\pm 0.001}$	$64.61_{\pm 0.20}$	$1.390_{\pm 0.008}$	$0.107_{\pm 0.001}$	$49.62_{\pm 0.15}$	$2.204_{\pm 0.003}$	$0.099_{\pm 0.002}$
	Laplace	$91.14_{\pm 0.12}$	$0.312_{\pm 0.005}$	$0.039_{\pm 0.001}$	$64.85_{\pm 0.13}$	$1.359_{\pm 0.006}$	$0.096_{\pm 0.001}$	$49.73_{\pm 0.18}$	$2.184_{\pm 0.007}$	$0.089_{\pm 0.001}$
Affine (Alg. 3)	Uniform	$91.60_{\pm 0.05}$	$0.300_{\pm 0.002}$	$0.040_{\pm 0.001}$	$66.08_{\pm 0.12}$	$1.288_{\pm 0.007}$	$0.093_{\pm 0.002}$	$51.19_{\pm 0.12}$	$2.099_{\pm 0.005}$	$0.076_{\pm 0.001}$
	Gaussian	$91.53_{\pm 0.10}$	$0.294_{\pm 0.004}$	$0.036_{\pm 0.001}$	$66.55_{\pm 0.10}$	$1.255_{\pm 0.005}$	$0.079_{\pm 0.002}$	$51.13_{\pm 0.16}$	$2.098_{\pm 0.004}$	$0.070_{\pm 0.002}$
	Laplace	$91.87_{\pm 0.04}$	$0.272_{\pm 0.002}$	$0.029_{\pm 0.001}$	$66.44_{\pm 0.10}$	$1.247_{\pm 0.006}$	$0.071_{\pm 0.001}$	$51.36_{\pm 0.14}$	$2.101_{\pm 0.009}$	$0.065_{\pm 0.001}$
SGD	–	$91.22_{\pm 0.07}$	$0.354_{\pm 0.006}$	$0.050_{\pm 0.001}$	$64.19_{\pm 0.14}$	$1.431_{\pm 0.007}$	$0.121_{\pm 0.001}$	$49.48_{\pm 0.10}$	$2.231_{\pm 0.004}$	$0.106_{\pm 0.001}$
iVON	Gaussian	$91.80_{\pm 0.05}$	$0.288_{\pm 0.003}$	$0.038_{\pm 0.001}$	$66.59_{\pm 0.11}$	$1.209_{\pm 0.004}$	$0.049_{\pm 0.001}$	$52.13_{\pm 0.08}$	$1.982_{\pm 0.003}$	$0.018_{\pm 0.001}$
VOGN	Gaussian	$91.32_{\pm 0.09}$	$0.264_{\pm 0.003}$	$0.011_{\pm 0.000}$	$66.81_{\pm 0.14}$	$1.183_{\pm 0.007}$	$0.020_{\pm 0.002}$	$51.09_{\pm 0.12}$	$2.045_{\pm 0.006}$	$0.016_{\pm 0.001}$

Table 2: Our proposed affine learning rule (Alg. 3) performs competitively to SGD and state-of-the art Bayesian approaches **VOGN** (Osawa et al., 2019) and the Adam-like optimizer from Lin et al. (2020, Figure 1) which we refer to as **iVON**. Empirically, making the q_0 distribution more heavy tailed (going from uniform to Gaussian, and from Gaussian to Laplace) improves calibration measures like ECE and NLL.

can be explained as the effect of entropy and the mean of a distribution being intimately tied together. Weight distributions of connections with large mean also are spread out, i.e., have large entropy. Therefore in avoiding large expected errors, any unnecessary non-robust weight magnitude is suppressed. We may interpret the resulting sharpness of the filters as neuronal task specialization in attributes we humans can convey such as color, location and orientation. For example, the readers probably can locate the multiplicative filters in Fig. 3 when referred to simply as "*the blue dot in the bottom right*" or "*multicolor vertical stripe left of center*". For the representations learned by the additive rule, no such short descriptions exist.

5.2 The affine learning rule

Finally, we compare our affine learning update from Sec. 4.3 to state-of-the-art natural-gradient variational inference methods: VOGN (Osawa et al., 2019) and the Adam-like optimizer given in Lin et al. (2020, Figure 1) which we refer to as iVON. For a detailed pseudo-code of our affine update rule, see Alg. 3 in the appendix. The comparison is carried out for a standard ResNet-20 architecture which reaches around 91% when trained with SGD, see (He et al., 2016, Table 6). The hyperparameters and other details are in App. B.2.

Table 2 summarizes our results. Our algorithm yields competitive results to SGD, VOGN and iVON, yet offers more flexibility in the choice of distribution: VOGN and iVON are updating a Gaussian distribution, whereas our method works for any base distribution q_0. Using a heavy-tailed Laplace distribution leads to improvements in NLL and ECE compared to a Gaussian or the even more thin-tailed uniform distribution. Moreover, both VOGN and iVON require a small additional damping term to stabilize the learning algorithm, see Osawa et al. (2019). Our algorithm does not require any such additional term, making it easier to tune.

The learning update rule arising from an additive group (Sec. 4.1) has been recently studied by Orvieto et al. (2022) in the context of regularizing noise injections. Using an affine update allows one to learn the variance of the noise. Table 2 shows that this leads to improvements while eliminating an additional hyperparameter which controls the strength of the noise. Additive and affine columns were computed with 1 MC sample only, thus the compute cost is comparable to that of SGD.

6 DISCUSSION

In this work we highlight the central role of Lie group actions in learning a distribution for the generalized Bayesian objective (1). The rich structures provided by the Lie group gives us flexibility in the choice of distributions, allows us to compute gradients via reparametrization tricks, lets us avoid recomputations of the Fisher metric and automatically satisfies the manifold constraints.

We have seen how different groups can lead to distinct learning behaviours, for example, multiplicative updates lead to sparse and localized features. The update-rule obtained by the affine group is a new adaptive learning rate algorithm, which has no overhead over SGD but learns a distribution which gives improved calibration results. The recipe we provide in this work can be used to design new learning algorithms for exploring similar intriguing behaviours.

Acknowledgements

We would like to thank Koichi Tojo, Akiyoshi Sannai, Asuka Takatsu, Benoît Collins and Kenichi Bannai for various discussions and feedback. This work was supported by the Bayes-duality project, JST CREST Grant Number JPMJCR2112. Eren Mehmet Kıral was supported by the RIKEN Special Postdoctoral Researcher Program.

Author Contributions Statement

List of Authors: Eren Mehmet Kıral (EMK), Thomas Möllenhoff (TM), Mohammad Emtiyaz Khan (MEK).

Based on in-depth discussions with TM, EMK proposed the Lie group framework, derived the specific algorithms and the connections to existing methods. MEK provided feedback on these. TM designed and conducted the experiments with suggestions from EMK and MEK. MEK and EMK wrote the paper together, with feedback from TM.

References

Amit, D. J., Wong, K. Y. M., and Campbell, C. (1989). Perceptron learning with sign-constrained weights. *Journal of Physics A: Mathematical and General*, 22(12):2039. 7

Baba, N. (1981). Convergence of a random optimization method for constrained optimization problems. *J. Optim. Theory Appl.*, 33(4):451–461. 2

Barndorff-Nielsen, O., Blæsild, P., and Eriksen, P. (2012). *Decomposition and invariance of measures, and statistical transformation models*, volume 58. Lecture Notes in Statistics, Springer Science & Business Media. 1, 4

Barndorff-Nielsen, O., Blæsild, P., Jensen, J. L., and Jørgensen, B. (1982). Exponential transformation models. *Proc. Roy. Soc. London Ser. A*, 379(1776):41–65. 1

Bernstein, J., Zhao, J., Meister, M., Liu, M.-Y., Anandkumar, A., and Yue, Y. (2020). Learning compositional functions via multiplicative weight updates. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2, 7

Beyer, H.-G. (2001). *The theory of evolution strategies*. Springer Science & Business Media. 2

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., and Kolesnikov, A. (2021). Knowledge distillation: A good teacher is patient and consistent. *arXiv:2106.05237*. 7

Catoni, O. (2007). *PAC-Bayesian Supervised Classification*. Number 56. Institute of Mathematical Statistics Lecture Notes – Monograph Series. 2

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 6(6):721–741. 2

Ghai, U., Hazan, E., and Singer, Y. (2020). Exponentiated gradient meets gradient descent. In Kontorovich, A. and Neu, G., editors, *Algorithmic Learning Theory, ALT 2020, 8-11 February 2020, San Diego, CA, USA*, volume 117 of *Proceedings of Machine Learning Research*, pages 386–407. PMLR. 7

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*. 8

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 9, 22

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106. 2, 6

Khan, M. E. and Lin, W. (2017). Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 3

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In *International Conference on Machine Learning (ICML)*. 1, 3

Khan, M. E. and Rue, H. (2021). The Bayesian learning rule. arXiv:2107.04562. 1, 2, 3, 6, 7, 8, 18, 21, 22

Lee, J. M. (2013). *Introduction to Smooth Manifolds*. Springer Science & Business Media. 3

Leordeanu, M. and Hebert, M. (2008). Smoothing-based optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2

Lin, W., Khan, M. E., and Schmidt, M. (2019a). Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning (ICML)*. 1, 3

Lin, W., Khan, M. E., and Schmidt, M. (2019b). Stein's lemma for the reparameterization trick with exponential family mixtures. arXiv:1910.13398. 1, 3

Lin, W., Nielsen, F., Khan, M. E., and Schmidt, M. (2021). Tractable structured natural gradient descent using local parameterizations. arXiv:2102.07405. 3

Lin, W., Schmidt, M., and Khan, M. E. (2020). Handling the positive-definite constraint in the Bayesian learning rule. In *International Conference on Machine Learning (ICML)*. 1, 3, 9, 21, 22

Meng, X., Bachmann, R., and Khan, M. E. (2020). Training binary neural networks using the Bayesian learning rule. In *International Conference on Machine Learning (ICML)*. 1, 3

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*. 2

Möllenhoff, T. and Khan, M. E. (2023). SAM as an optimal relaxation of Bayes. In *International Conference on Learning Representations (ICLR)*. 1, 3

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609. 2, 6

Orvieto, A., Kersting, H., Proske, F., Bach, F., and Lucchi, A. (2022). Anticorrelated noise injection for improved generalization. In *International Conference on Machine Learning (ICML)*. 6, 9

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., and Khan, M. E. (2019). Practical deep learning with Bayesian principles. *Advances in Neural Information Processing Systems (NeurIPS)*. 1, 3, 9, 22

Salimans, T. and Knowles, D. A. (2013). Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882. 3

Spall, J. (2005). *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons. 2

Whittington, J., Dorrell, W., Ganguli, S., and Behrens, T. (2022). Disentangling with biological constraints: A theory of functional cell types. arXiv:2210.01768. 2, 7

Williams, R. J. and Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268. 2

Zellner, A. (1988). Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280. 2

A MATHEMATICAL DETAILS

A.1 The differentials and the Fisher metric

A.1.1 Fisher is the second differential of KL-Divergence

The Fisher metric can be obtained as the second order differential of our objective function $\mathcal{E}(q) = \mathbb{D}(q \| p_\tau)$. Let $h \in T_q\mathcal{Q}$ be a tangent vector and let us perturb the energy functional by a mean zero function h. Its second order approximation is given by

$$\mathcal{E}(q+h) \approx \mathcal{E}(q) + (\mathrm{d}\mathcal{E}(q))[h] + \frac{1}{2}(\mathrm{d}^2\mathcal{E}(q))[h,h]. \tag{18}$$

Let us give a sketch of calculation the terms in this quadratic expansion of $\mathcal{E}$, only giving the main idea, as the derivation of the second differential of KL-divergence is already well known, see, e.g. John Baez's blog `https://math.ucr.edu/home/baez/information/information_geometry_7.html` (as of Oct 11, 2022). Perturb by h,

$$\begin{aligned}\mathcal{E}(q+h) = \mathbb{D}(q+h \| p_\tau) &= \int (q+h)\log\left(\frac{q+h}{p_\tau}\right) \\ &= \int (q+h)\log\left(\frac{q}{p_\tau}\right) + \int (q+h)\log\left(\frac{q+h}{q}\right) \\ &= \underbrace{\int q\log\left(\frac{q}{p_\tau}\right)}_{\mathcal{E}(q)} + \int h\log\left(\frac{q}{p_\tau}\right) + \int (q+h)\left(\frac{h}{q} - \frac{h^2}{q^2} + \cdots\right),\end{aligned}$$

where in the second line we multiplied and divided the ratio inside the logarithm by q and in the last line we applied the Taylor expansion of $\log(1+x)$ with $x = h/q$. Continuing the calculation we get that

$$\mathcal{E}(q+h) - \mathcal{E}(q) = \int h\log\left(\frac{q}{p_\tau}\right) + \cancelto{0}{\int h} + \frac{1}{2}\int \frac{h^2}{q} + \cdots$$

The linear term must be the first differential and the quadratic term is the second differential $(\mathrm{d}^2\mathcal{E}(q))[h,h]$. Using polarization identities we can get the quadratic term as a bilinear form

$$(\mathrm{d}\mathcal{E}(q))[h] = \int_\Theta h(\theta)\log\left(\frac{q(\theta)}{p_\tau(\theta)}\right)\mathrm{d}\theta \qquad \text{and} \qquad (\mathrm{d}^2\mathcal{E})(q))[h_1,h_2] = \int_\Theta \frac{h_1(\theta)}{q(\theta)}\frac{h_2(\theta)}{q(\theta)}q(\theta)\,\mathrm{d}\theta. \tag{19}$$

The second differential can also be written in the form $\mathbb{E}_q[(\partial_{h_1}\log q)(\partial_{h_2}\log q)]$, this is exactly the Fisher metric. Therefore one can see that choosing the Fisher metric as the direction of fastest descent is also compatible with minimizing the quadratic expansion of the objective function $\mathcal{E}$.

A.1.2 Fisher metric is independent of base point

If the tangent vectors are parametrized by the Lie algebra as h_X then we can write them as $\pi(g)$ of a tangent vector at identity:

$$h_X = \frac{\mathrm{d}}{\mathrm{d}t}\pi(ge^{tX})q_0\bigg|_{t=0} = \pi(g)\frac{\mathrm{d}}{\mathrm{d}t}q_{e^{tX}}\bigg|_{t=0} = \pi(g)h_X^e.$$

This means that by making a change of variables $\theta \mapsto g\cdot\theta$ the we get a quantity that is independent of g. Indeed

$$\begin{aligned}\omega_{\text{Fisher}}(h_X^g, h_Y^g) &= \int_\Theta \frac{h_X^g(\theta)}{q_g(\theta)}\frac{h_Y^g(\theta)}{q_g(\theta)}q_g(\theta)\,\mathrm{d}\theta = \int_\Theta \frac{h_X^e(g^{-1}\cdot\theta)}{q_0(g^{-1}\cdot\theta)}\frac{h_Y^e(g^{-1}\cdot\theta)}{q_0(g^{-1}\cdot\theta)}\left|\frac{\mathrm{d}(g\cdot\theta)}{d\theta}\right|^{-1} q_0(g^{-1}\cdot\theta)\,\mathrm{d}\theta \\ &= \int_\Theta \frac{h_X^e(\theta)}{q_0(\theta)}\frac{h_Y^e(\theta)}{q_0(\theta)}q_0(\theta)\,\mathrm{d}\theta = \omega_{\text{Fisher}}(h_X^e, h_Y^e).\end{aligned}$$

for any $X, Y \in T_eG$ and $g \in G$. In the second line we made use of a change of variables $\theta \mapsto g\cdot\theta$. This is simply a bilinear form in the Lie algebra T_eG. So, it is enough to compute the Fisher metric once and for all, and we do not need to compute a different metric at each point throughout the training.

A.2 Calculation of $\mathrm{d}\mathcal{E}(q)$

For the first differential given the form of the Gibbs distribution $p_\tau(\theta) \propto e^{-\frac{1}{\tau}\ell(\theta)}$ and making use of $\int_\Theta h(\theta)\,\mathrm{d}\theta = 0$ we get the formula (7).

We start with (7)

$$(\mathrm{d}\mathcal{E}(q_g))[h] = \frac{1}{\tau}\underbrace{\int h(\theta)\ell(\theta)\,\mathrm{d}\theta}_{①} + \underbrace{\int h(\theta)\log q_g(\theta)\,\mathrm{d}\theta}_{②}. \tag{20}$$

The first term ① having the dependence on the loss function turns into

$$\begin{aligned}
① &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int q_{ge^{tX}}(\theta)\ell(\theta)\,\mathrm{d}\theta\right)\Bigg|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int \frac{1}{|\,\mathrm{d}(ge^{tX}\cdot\theta)/\,\mathrm{d}\theta|} q_0(e^{-tX}g^{-1}\cdot\theta)\ell(\theta)\,\mathrm{d}\theta\right)\Bigg|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int q_0(\theta)\ell\left(ge^{tX}\cdot\theta\right)\,\mathrm{d}\theta\right)\Bigg|_{t=0} \\
&= \int q_0(\theta)(\nabla_\theta\ell(g\cdot\theta))^\top \frac{\mathrm{d}}{\mathrm{d}t}\left(ge^{tX}\cdot\theta\right)\Bigg|_{t=0}\mathrm{d}\theta
\end{aligned} \tag{21}$$

We have used the definition of $h = h_X$ and differentiating under the integral sign in the first line, the definition of q_g in the second line, and a change of variables $\theta \mapsto ge^{tX}\cdot\theta$ in the third line. After this change of variables note that t is now in ℓ and a derivative in t will introduce $\nabla_\theta\ell$ via the chain rule.

We can go back to writing thing in terms of expectations over q_g by applying a final change of variables

$$\begin{aligned}
① &= \int q_g(\theta)\left(\nabla_\theta\ell(\theta)\right)^\top \frac{\mathrm{d}}{\mathrm{d}t}\left(ge^{tX}g^{-1}\cdot\theta\right)\Bigg|_{t=0}\mathrm{d}\theta \\
&= \int q_g(\theta)\left(\nabla_\theta\ell(\theta)\right)^\top \frac{\mathrm{d}}{\mathrm{d}t}\left(ge^{tX}g^{-1}\right)\Bigg|_{t=0}\cdot\theta\,\mathrm{d}\theta \\
&= \int q_g(\theta)\left(\nabla_\theta\ell(\theta)\right)^\top \mathrm{Ad}_g(X)\cdot\theta\,\mathrm{d}\theta
\end{aligned} \tag{22}$$

Notice the appearance of the adjoint representation Ad_g, the derivative is exactly $\mathrm{Ad}_g(X)\cdot\theta$ where $\mathrm{Ad}_g : T_eG \to T_eG$ is the linear operator defined as $\mathrm{Ad}_g(X) = \frac{\mathrm{d}}{\mathrm{d}t}ge^{tX}g^{-1}\big|_{t=0} \in T_eG$.

The entropic contribution ② to the differential can also be calculated similarly

$$\begin{aligned}
② &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int q_{ge^{tX}}(\theta)\log q_g(\theta)\,\mathrm{d}\theta\right)\Bigg|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int q_0(\theta)\log\left(\frac{1}{|\,\mathrm{d}(g\cdot\theta)/\,\mathrm{d}\theta|}q_0\left(e^{tX}\cdot\theta\right)\right)\,\mathrm{d}\theta\right)\Bigg|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\Bigg(\underbrace{-\int\log(|\,\mathrm{d}(g\cdot\theta)/\,\mathrm{d}\theta|)q_0(\theta)}_{\to 0} + \int q_0(\theta)\log q_0\left(e^{tX}\cdot\theta\right)\,\mathrm{d}\theta\Bigg)\Bigg|_{t=0} \\
&= \int\left(\nabla_\theta q_0(\theta)\right)^\top\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{tX}\cdot\theta\right)\Bigg|_{t=0}\mathrm{d}\theta = \int\left(\nabla_\theta q_0(\theta)\right)^\top X\cdot\theta\,\mathrm{d}\theta.
\end{aligned} \tag{23}$$

On the first line we use the definition of the tangent vector h_X, the second line makes a change of variables $\theta \mapsto ge^{tX}\cdot\theta$. We can cancel the $|\,\mathrm{d}(g\cdot\theta)/\,\mathrm{d}\theta|$ term in the third line because it has no t dependence, and on the last line we go one step further and apply the chain rule, cancelling the q_0 term via the logarithmic derivative. The definition of the action of T_eG on Θ is given exactly as the linearization i.e. $X\cdot\theta := \frac{\mathrm{d}}{\mathrm{d}t}e^{tX}\cdot\theta\big|_{t=0}$.

A.3 Specializing the update rule to the additive group

Algorithm 1 Additive Update

Input: $\alpha > 0$ step size, $K \geq 1$ MC-sample number, $\beta \in [0, 1)$, ν a distribution on $\mathbb{R}$.
Initialize $g \in \mathbb{R}^P$ randomly, and $M = \mathbf{0} \in \mathbb{R}^P$.
while not converged **do**
  Sample a minibatch $\mathcal{J} \subseteq [1..N]$ of size n,
  Sample noise vectors $\varepsilon_k \sim \nu^P$ for $k = 1, \dots, K$,
  Put $\theta_k = g + \varepsilon_k$, the MC-parameter samples,
  $U \leftarrow \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{n} \sum_{j \in \mathcal{J}} \nabla \ell_j(\theta_k) + \frac{1}{N} \nabla R(\theta_k) \right)$,
  $M \leftarrow (1 - \beta) U + \beta M$,
  $g \leftarrow g - \alpha M$,
end while
Return g.
Loss function: $\ell(\theta) = \sum_{i=1}^{N} \ell_i(\theta) + R(\theta)$.

In this case $\Theta = \mathbb{R}^P$ with $G = (\mathbb{R}^P, +)$ acting by $g \cdot \theta = \theta + g$ on the space of parameters Θ. The tangent bundle of G is trivial, with each tangent space isomorphic to $\mathbb{R}^P$. The exponential map for $X \in T_g(\mathbb{R}^P) \cong \mathbb{R}^P$ is simply given by identity, i.e. $\exp(X) = X$.

We work in the mean-field case. That is, let $\tilde{q}_0$ be the density function of an everywhere positive probability distribution on $\mathbb{R}$, and put $q_0(\theta) = \prod_i \tilde{q}_0(\theta_i)$. By making this choice, we assume that the probability for each parameter is independent and identically distributed. Taking the base point distribution q_0 the orbit of q_0 under under the action of G via pushforwards gives us $\mathcal{Q} := \{q_g(\theta)\,\mathrm{d}\theta : g \in \mathbb{R}^P\}$ where $q_g(\theta) := (\pi(g) \cdot q_0)(\theta) = q_0(\theta - g)$.

The infinitesimal action of $T_e G$ on Θ is given as $X \cdot \theta = \frac{\mathrm{d}}{\mathrm{d}t}(\exp(tX) \cdot \theta)\big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}(\theta + tX)\big|_{t=0} = X \in T_\theta \Theta$. And since the group is commutative, the adjoint representation is trivial, i.e. $\mathrm{Ad}_g(X) \cdot \theta = X \cdot \theta = X$.

The differential of E can be calculated by (7). First note that the integral $\textcircled{2}$ in (9) vanishes as we get $\textcircled{2} = \left(\int \nabla_\theta q_0(\theta)\,\mathrm{d}\theta \right)^\top X = \mathbf{0}^\top X = 0$. Indeed upon integration by parts

$$\int_\Theta \partial_i q_0(\theta)\,\mathrm{d}\theta = \int_{\mathbb{R}} \tilde{q}_0'(\theta_i)\,\mathrm{d}\theta_i \times \prod_{j \neq i} \int_{\mathbb{R}} \tilde{q}_0(\theta_j)\,\mathrm{d}\theta_j = \tilde{q}_0(\theta_i)\Big|_{\theta_i = -\infty}^{\infty} = 0 - 0 = 0.$$

This fact should not be surprising since the group action only translates the mean of the distribution q_0 and there is no change in entropy. As for (8) we may again take the X dependence out and write the integral as an expectation. Thus the differential is calculated as

$$(\mathrm{d}\mathcal{E}(q_g))[h_X] = \textcircled{1} = \mathbb{E}_{q_g}[\nabla_\theta \ell]^\top X.$$

Since $\mathrm{Ad}_g(X) \cdot \theta = X \cdot \theta = X$. Now we calculate the Fisher information matrix, to so that we may apply the musical-isomorphism $\sharp$ and get an element of $T_{q_g}\mathcal{Q}$ instead of the covector $\mathrm{d}\mathcal{E}(q_g) \in T^*_{q_g}\mathcal{Q}$.

The tangent vectors to $\mathcal{Q}$ are given as mean-zero functions. More concretely in our case at any $q = q_g$ the tangent space is spanned by the

$$h_i(\theta) = (\partial_i q_0)(\theta - g)$$

and with respect to this basis the Fisher information metric is calculated as,

$$\omega_{\text{Fisher}}(h_i, h_j) = \int_{\mathbb{R}^P} \frac{\partial_i q_0(\theta)}{q_0(\theta)} \frac{\partial_j q_0(\theta)}{q_0(\theta)} q_0(\theta)\,\mathrm{d}\theta = \int_{\mathbb{R}^P} \frac{\tilde{q}_0'(\theta_i)}{\tilde{q}_0(\theta_i)} \frac{\tilde{q}_0'(\theta_j)}{\tilde{q}_0(\theta_j)} q_0(\theta)\,\mathrm{d}\theta.$$

For $i \neq j$ this reduces to $\left(\int_{\mathbb{R}} \tilde{q}_0'(\theta)\,\mathrm{d}\theta \right)^2 = 0$, and so F is given by a scalar matrix $F = c_F I_{P \times P}$ where the constant $c_F > 0$ is given by the integral

$$c_F = \int_{\mathbb{R}} \frac{\tilde{q}_0'(\theta)^2}{\tilde{q}_0(\theta)}\,\mathrm{d}\theta.$$

The update rule in this case is then given by

$$g^{\text{updated}} = g + \exp\left(-\alpha \mathbb{E}_{q_g}[\nabla_\theta \ell] \right) = g - \alpha \mathbb{E}_{q_g}[\nabla_\theta \ell] \tag{24}$$

for some step size $\alpha > 0$, where we have absorbed the Fisher constant c_F into the step size. Note that absorbing the Fisher constant into the step-size is exactly why we are able to then substitute distributions such as Dirac delta as q_0.

A.4 Specializing to the multiplicative group

Algorithm 2 Multiplicative Update

Input: $\alpha > 0$ step size, $K \geq 1$ MC-sample number, T temperature, $\beta \in [0, 1)$, ν a distribution on $\mathbb{R}_{>0}$.

Initialize $g \in \mathbb{R}^P_{>0}$ randomly, and $M = \mathbf{0} \in \mathbb{R}^P$.
while not converged **do**
  Sample a minibatch $\mathcal{J} \subseteq [1..N]$ of size n,
  Sample noise vectors $\varepsilon_k \sim \nu^P$ for $k = 1, \ldots, K$,
  Put $\theta_k = g\varepsilon_k$, the MC-parameter samples,
  $U_k = \frac{1}{n} \sum_{j \in \mathcal{J}} \theta_k \nabla \ell_j(\theta_k) + \frac{1}{N} \theta_k \nabla R(\theta_k) - \frac{\tau}{N} \mathbf{1}$,
  $U \leftarrow \frac{1}{K} \sum_{k=1}^{K} U_k$
  $M \leftarrow (1 - \beta) U + \beta M$,
  $g \leftarrow g \exp(-\alpha M)$
end while
Return g.

Loss function: $\ell(\theta) = \sum_{i=1}^{N} \ell_i(\theta) + R(\theta)$. *Here,* ab *for two vectors* $a, b \in \mathbb{R}^P$ *is taken to mean componentwise multiplication and* exp *is also computed componentwise.*

For the multiplicative case we have $\Theta = \mathbb{R}^P_{>0}$. Fixing the signs of a neural network's weights we obtain such a model, where the (positive) magnitudes of the weights become the parameters of the model. The group $G = \mathbb{R}^P_{>0}$, considered with componentwise multiplication as the group operation acts on Θ in the same way $g \cdot \theta := g\theta$. Here the componentwise product of two vectors a, b is simply denoted by ab.

Again at a mean-field base distribution $q_0(\theta) = \prod_{k=1}^{P} \tilde{q}_0(\theta_k)$, the orbit of the pushforward measures look $\mathcal{Q} = \{q_g = \pi(g) \cdot q_0 : g \in G\} = \{\frac{1}{|g|} q_0(g^{-1}\theta) : g \in G\}$. Here $|g| = \prod_{k=1}^{P} g_i$ is the determinant of the Jacobian $|\,\mathrm{d}(g \cdot \theta)/\,\mathrm{d}\theta|$.

A.4.1 Examples

This scheme includes important families.

- Choose $\widetilde{q}_0(\theta) = e^{-\theta}$, then we get the family of exponential distributions
$$\pi(g) \cdot q_0(\theta) = \prod_k \frac{1}{g_k} e^{-\theta_k / g_k}$$
is the family of exponential distributions $\{\prod_i \lambda_i e^{-\langle \theta, \lambda \rangle} : \lambda \in \mathbb{R}^P_{>0}\}$. The group parameter g and the natural parameter λ are componentwise reciprocals of each other.
- Choosing $\widetilde{q}_0(\theta) = \theta e^{-\frac{1}{2}\theta^2}$ gives us the family of Rayleigh distributions
$$\pi(g) \cdot q_0(\theta) = \prod_k \frac{\theta_i}{g_i^2} e^{-\theta_i^2/(2g_i^2)}.$$
In this case the σ_i parameter of the Rayleigh distribution exactly match up with g_i.
- Log-normal distributions with a fixed variance parameter also fit into this family scheme. Let us put
$$\widetilde{q}_0(\theta) = \frac{1}{\theta\sigma\sqrt{2\pi}} e^{-\frac{(\ln\theta - m)^2}{2\sigma^2}}$$
Then if $g > 0$ we get
$$\frac{1}{g}\widetilde{q}_0(g^{-1}\theta) = \frac{1}{\theta\sigma\sqrt{2\pi}} e^{-\frac{(\ln\theta - \ln g - m)^2}{2\sigma^2}},$$
the log-normal distribution with mean $m + \ln g$ and the same scale parameter σ. The action of $\pi(g)$ translates the parameter m by $\ln g$.

A.4.2 The tangent vectors

The Lie algebra of G is given by vectors $X \in \mathbb{R}^P$. Using that we can parametrize the tangent space of $\mathcal{Q}$ at any q_g.

Lemma 1. *Given a $q = q_g \in \mathcal{Q}$ we have a basis of tangent vectors given as functions on θ. They integrate to 0, and are explicitly given by*

$$h_i(\theta) = -\pi(g) \cdot \left(q_0(\theta) \left(1 + \theta_i \frac{\tilde{q}_0'(\theta_i)}{\tilde{q}_0(\theta_i)} \right) \right).$$

Proof. We calculate from the definition.

$$\begin{aligned} h_X^g(\theta) &= \frac{\mathrm{d}}{\mathrm{d}t} q_{g\exp(tX))}(\theta) \Big|_{t=0} \\ &= \frac{d}{\mathrm{d}t} \left(\frac{1}{|g\exp(tX)|} q_0(\exp(-tX)g^{-1} \cdot \theta) \right) \Bigg|_{t=0} \\ &= -\frac{(\sum_i X_i)}{|g|} q_0(g^{-1}\theta) - \sum_i \frac{1}{|g|} \partial_i q_0(g^{-1}\theta) X_i \frac{\theta_i}{g_i}. \end{aligned}$$

Here exp means that we apply exponentiation componentwise, as well as product of two vectors. Substituting the standard basis e_i for X gives a basis for $T_{q_g}\mathcal{Q}$. Let us call it $h_1, h_2, \dots, h_P$. The above formula can then be written more succinctly as

$$h_i(\theta) = -q_g(\theta) \left(1 + \frac{\theta_i}{g_i} \frac{\partial_i q_0(g^{-1}\theta)}{q_0(g^{-1}\theta)} \right) = -q_g(\theta) \left(1 + \frac{\theta_i}{g_i} \frac{\tilde{q}_0'(\frac{\theta_i}{g_i})}{\tilde{q}_0(\frac{\theta_i}{g_i})} \right).$$

Noting the implicit $\pi(g)$ action, we get the result. □

Another way to write these tangent vectors are $h_i(\theta) = -q_g(\theta) - \pi(g)\delta_i q_0(\theta)$ where δ_i are the invariant differential operators on functions on Θ given by $\delta_i f(\theta) = (\theta_i \partial_i f)(\theta)$. The reason this is called an *invariant* operator is because it is invariant under the group action by G, in other words $\delta_i \pi(g) f = \pi(g)\delta_i f$ for all $i = 1, \dots, P$.

A.4.3 The Fisher metric

We now calculate the Fisher metric as a matrix with respect to the basis of tangent vectors given above in Lemma 1.

Lemma 2. *The matrix for the Fisher bilinear form with respect to the given basis $h_1, \dots, h_P \in T_{q_g}\mathcal{Q}$ above, is $c_F I$. Here $I = I_{P\times P}$ is the identity matrix and $c_F > 0$ is a constant that depends only on $\tilde{q}_0$.*

Notice that this metric does not depend on g, i.e. with this parametrization it is independent of the basepoint q_g.

Proof. The Fisher bilinear form is $\int (\partial_{h_i} \log q_g(\theta))(\partial_{h_j} \log q_g(\theta)) q_g(\theta)\, \mathrm{d}\theta$ by definition. Here ∂_h means that we are taking a directional derivative in the space of all measures in the direction of h. We calculate,

$$\begin{aligned} \omega_{\text{Fisher}}(h_i, h_j) &= \int \frac{h_i(\theta)}{q_g(\theta)} \frac{h_j(\theta)}{q_g(\theta)} q_g(\theta)\, \mathrm{d}\theta \\ &= \int \left(1 + \frac{\theta_i}{g_i} \frac{\tilde{q}_0'(\frac{\theta_i}{g_i})}{\tilde{q}_0(\frac{\theta_i}{g_i})} \right) \left(1 + \frac{\theta_j}{g_j} \frac{\tilde{q}_0'(\frac{\theta_j}{g_j})}{\tilde{q}_0(\frac{\theta_j}{g_j})} \right) q_g(\theta)\, \mathrm{d}\theta \\ &= \int \left(1 + \theta_i \frac{\tilde{q}_0'(\theta_i)}{\tilde{q}_0(\theta_i)} \right) \left(1 + \theta_j \frac{\tilde{q}_0'(\theta_j)}{\tilde{q}_0(\theta_j)} \right) q_0(\theta)\, \mathrm{d}\theta. \end{aligned}$$

In the last line we made a change of variables $\theta \mapsto g \cdot \theta$. Now there are two cases, firstly if $i \neq j$,

$$\begin{aligned} \omega_{\text{Fisher}}(h_i, h_j) &= \prod_{k \neq i,j} \int_0^\infty \tilde{q}_0(\theta_k)\, \mathrm{d}\theta_k \left(\int_0^\infty (\tilde{q}_0(\theta_i) + \theta_i \tilde{q}_0'(\theta_i))\, \mathrm{d}\theta_i \right) \left(\int_0^\infty (\tilde{q}_0(\theta_j) + \theta_j \tilde{q}_0'(\theta_j))\, \mathrm{d}\theta_j \right) \\ &= \left(\int_0^\infty \frac{\mathrm{d}}{\mathrm{d}\theta} (\theta \tilde{q}_0(\theta))\, \mathrm{d}\theta \right)^2 = \left(\theta \tilde{q}_0(\theta) \Big|_{t=0}^\infty \right)^2 = 0. \end{aligned}$$

Thus with respect to this basis, the matrix of the Fisher metric is diagonal. The value at these diagonal elements is calculated via

$$c_F := \int_0^\infty \left(1 + \theta \frac{\tilde{q}_0'(\theta)}{\tilde{q}_0(\theta)}\right)^2 \tilde{q}_0(\theta)\,\mathrm{d}\theta.$$

This is a nonnegative number. Most importantly it is independent of g and only depends on the $\tilde{q}_0$ we chose. □

The constant of this lemma is given as $c_F = 1$ if $\widetilde{q}_0(\theta) = e^{-\theta}$ giving the exponential distributions, and $c_F = 4$ if $\widetilde{q}_0(\theta) = \theta e^{-\frac{\theta^2}{2}}$. In general c_F is clearly nonnegative, but above we claimed more. That it was positive. The only way the integral could be zero is if $\tilde{q}_0$ satisfies the differential equation $\tilde{q}_0'(\theta) = -\tilde{q}_0(\theta)/\theta$ which has solutions $\tilde{q}_0 = c/\theta$. Notice that these solutions do not have finite integrals on $(0, \infty)$ and thus fall outside our purview.

A.4.4 The Differential of $\mathcal{E}$

We apply (7) and the calculations below that to our specific situation.

Lemma 3. *Let $h_X = \sum_i X_i h_i$ be a tangent vector in $T_{q_g}\mathcal{Q}$.Then the differential of $\mathcal{E}$ at q_g evaluated at h_X is given as*

$$(\mathrm{d}\mathcal{E}(q_g))[h_X] = n^\top X \quad \textit{where} \quad n = \frac{1}{\tau}\mathbb{E}_{q_g}\left[\theta \nabla_\theta \ell\right] - \mathbf{1}.$$

Here $\mathbf{1} = \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^\top$.

Proof. Note that $T_eG \cong \mathbb{R}^P$ and exponential map is componentwise exponentiation: $e^X = (e^{X_1}, e^{X_2}, \ldots, e^{X_P}) \in G$. Therefore the infinitesimal action of the Lie algebra on the parameters is given as $X \cdot \theta = \frac{\mathrm{d}}{\mathrm{d}t}(e^{tX} \cdot \theta)\big|_{t=0} = X\theta$ (again, understood as componentwise multiplication). Also as the group is abelian $\mathrm{Ad}_g(X) = X$. We use (22)

$$\textcircled{1} = \int q_g(\theta)(\nabla \ell(\theta))^\top (X\theta)\,\mathrm{d}\theta = \sum_{i=1}^P X_i \int q_g(\theta)\theta_i \partial_i \ell(\theta)\,\mathrm{d}\theta = \mathbb{E}_{q_g}[\theta \nabla_\theta \ell]^\top X$$

For the second part

$$\textcircled{2} = \sum_i X_i \int \theta_i (\partial_i q_0)(\theta) = \sum_i X_i \int_0^\infty \theta_i \tilde{q}_0'(\theta_i)\,\mathrm{d}\theta_i$$

where on the last equality we used the multiplicative structure of q_0. The resulting integral can be calculated via integration by parts as

$$\int_0^\infty \theta \tilde{q}_0'(\theta)\,\mathrm{d}\theta = \theta \tilde{q}_0(\theta)\Big|_0^\infty - \int_0^\infty \tilde{q}_0(\theta)\,\mathrm{d}\theta = -1.$$

The contribution from this part will be $-\mathbf{1}^\top X$. Therefore

$$(\mathrm{d}\mathcal{E}(q_g))[h_X] = \left(\frac{1}{\tau}\mathbb{E}_{q_g}[\theta \nabla_\theta \ell]^\top - \mathbf{1}\right)^\top X$$

as claimed. □

A.4.5 The update rule

We combine results of the previous section in order to write down the explicit update rule on $\mathcal{Q}$.

Firstly let us note that we are looking for a vector $n \in T_eG$ (for natural gradient) such that for any $X \in T_eG$ we have that

$$\omega^{Fisher}(h_n, h_X) = \mathrm{d}\mathcal{E}\big|_{q_g}[h_X].$$

By Lemma 3 and Lemma 2 we have $n = \frac{1}{c_F}\left(\frac{1}{\tau}\mathbb{E}_{q_g}\left[\theta \nabla_\theta \ell\right] - \mathbf{1}\right)$.

Thus we know that our direction of descent should be $-n$ and we can absorb $1/c_F$ and temperature τ into the step-size. Use the exponential map $\exp : T_eG \mapsto G$ followed by L_g as the retraction. Choosing $\alpha c_F \tau > 0$ as a step size, we have

$$g^{\text{updated}} = g\exp(-\alpha n) = g\exp\left(-\alpha\left(\mathbb{E}_{q_g}[\theta \nabla_\theta \ell] - \tau \mathbf{1}\right)\right).$$

Componentwise this reads as

$$g_i^{\text{updated}} = g_i \exp(-\alpha \left(\mathbb{E}_{q_g}\left[\theta_i \partial_i \ell\right] - \tau\right)). \tag{25}$$

This update rule naturally preserves the condition that $g_i > 0$, i.e. we stay on the manifold since the exponential map is not only defined locally but on all $T_e G$. This could not be guaranteed in the update rule by Khan and Rue (2021) which used an only locally defined retraction function.

A.5 Specializing to the diagonal affine group

Algorithm 3 Affine Update

Input: $\alpha > 0$ step size, $K \geq 1$ MC-sample number, $\beta_1, \beta_2 \in [0, 1)$, ν an even smooth distribution on $\mathbb{R}$.

Put $c_X = \int_{\mathbb{R}} \left(1 + \theta \frac{\nu'(\theta)}{\nu(\theta)}\right)^2 \mathrm{d}\theta$, $c_y = \int_{\mathbb{R}} \frac{\nu'(\theta)^2}{\nu(\theta)} \mathrm{d}\theta$.
Initialize $(A, b) \in \mathbb{R}^P_{>0} \times \mathbb{R}^P$ randomly, and $M_U = M_V = \mathbf{0} \in \mathbb{R}^P$.
while not converged **do**
  Sample a minibatch $\mathcal{J} \subseteq [1..N]$ of size n,
  Sample noise vectors $\varepsilon_k \sim \nu^P$ for $k = 1, \ldots, K$,
  Put $\theta_k = A\varepsilon_k + b$, the MC-parameter samples,
  $U_k = \frac{1}{n} \sum_{j \in \mathcal{J}} A\varepsilon_k \nabla \ell_j(\theta_k) + \frac{1}{N} A\varepsilon_k \nabla R(\theta_k) - \frac{\tau}{N}\mathbf{1}$,
  $U \leftarrow \frac{1}{c_X K} \sum_{k=1}^K U_k$,
  $V_k = \frac{1}{n} \sum_{j \in \mathcal{J}} A \nabla \ell_j(\theta_k) + \frac{1}{N} A \nabla R(\theta_k)$
  $V \leftarrow \frac{1}{c_y K} \sum_{k=1}^K V_k$
  $M_V \leftarrow (1 - \beta_1)V + \beta_1 M_V$,
  $M_U \leftarrow (1 - \beta_2)U + \beta_2 M_U$,
  $b \leftarrow b + A \frac{\exp(-\alpha M_U) - I}{M_U} M_V$
  $A \leftarrow A \exp(-\alpha M_U)$,
end while
Return (A, b).

Loss function: $\ell(\theta) = \sum_{i=1}^N \ell_i(\theta) + R(\theta)$. *Here, the product of two vectors is again taken to mean componentwise as well as the exponential. The function* $(e^{-\alpha x} - 1)/x$ *is well defined for all* x*, in evaluating it near* $x = 0$ *we may simply use the linear Taylor approximation in order to avoid division by* 0.

A.5.1 The Affine group and its action

The affine group combines the freedom of translations of the additive group and the scaling of the multiplicative groups. As with the above two groups we will use the mean field distribution, and therefore the scaling will be componentwise.

Realize the diagonal affine group $\mathrm{Aff}_P^{\mathrm{diag}}(\mathbb{R})$ as pairs (A, b) where A is a $P \times P$ positive diagonal matrix and $b \in \mathbb{R}^P$. The group operation is given as $(A_1, b_1)(A_2, b_2) = (A_1 A_2, A_1 b_2 + b_1)$. The action on parameters $\Theta = \mathbb{R}^P$ is

$$(A, b) \cdot \theta = A\theta + b. \tag{26}$$

This is compatible with the group multiplication defined above meaning $g_1 \cdot (g_2 \cdot \theta) = (g_1 g_2) \cdot \theta$, in fact this is why the group operation has been defined in such a way.

The group $\mathrm{Aff}_P^{\mathrm{diag}}(\mathbb{R})$ can be realized as a subgroup of matrices,

$$(A, b) \mapsto \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} \in \mathrm{GL}_{P+1}(\mathbb{R})$$

is an injective group homomorphism. This is a Lie group, with a Lie algebra consisting of pairs $\{(X, y) : X$ a diagonal $P \times P$ matrix, $y \in \mathbb{R}^P\}$ and the Lie bracket is given by

$$[(X_1, y_1), (X_2, y_2)] = (0, X_1 y_2 - X_2 y_1).$$

The exponential for this group is $\exp_e((X, y)) = (e^X, \frac{e^X - 1}{X} y)$, as can be seen most easily from the matrix representation of Lie algebra elements as $\left(\begin{smallmatrix} X & y \\ 0 & 0 \end{smallmatrix}\right)$. The meaning of $(e^X - I)/X$ is best understood interms of the Taylor expansion, and the expansion begins as $I + \frac{1}{2}X + \frac{1}{6}X^2 + \cdots$.

A.5.2 The information manifold $\mathcal{Q}$ and its geometry

Pick a base distribution $\tilde{q}_0 \in \mathcal{P}^+(\mathbb{R})$, and put $q_0(\theta) = \prod_{i=1}^P \tilde{q}_0(\theta_i) \in \mathcal{P}^+(\mathbb{R})$ and let $\mathcal{Q} = \{q_g(\theta) = \pi(g)q_0(\theta) : g \in G\} = \{\frac{1}{|A|}q_0(A^{-1}(\theta - b)) : (A, b) \in \mathrm{Aff}_P^{\mathrm{diag}}(\mathbb{R})\}$. Here $|A|$ denotes the absolute value of the determinant of A, which is the Jacobian determinant $|\,\mathrm{d}((A, b)\cdot\theta)/\,\mathrm{d}\theta)|$.

Lemma 4. *Let $q = q_g \in \mathcal{Q}$. The following $2P$ tangent vectors form a basis of $T_q\mathcal{Q}$:*

$$h_i^X = -\pi(g)\,(q_0 + \delta_i q_0) \qquad \textit{and} \qquad h_i^y = -\pi(g)\partial_i q_0.$$

for $i = 1, \ldots, P$. Here $(\delta_i f)(\theta) = \theta_i \partial_i f(\theta)$.

Proof. Given $(X, y) \in T_eG$ with $X = \mathrm{diag}((X_1, \ldots, X_P))$ we obtain a tangent vector as $\frac{\mathrm{d}}{\mathrm{d}t}q_{g e^{t(X,y)}}\big|_{t=0}$, call it $h_{(X,y)} = h^g_{(X,y)}$. Calculating explicitly,

$$\begin{aligned} h_{(X,y)}(\theta) &= \frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{|Ae^{tX}|}q_0\left(e^{-tX}A^{-1}\theta - e^{-tX}A^{-1}b + \tfrac{e^{-tX}-I}{tX}ty\right)\Bigg|_{t=0} \\ &= -\sum_i \frac{X_i}{|A|}q_0(A^{-1}(\theta - b)) - \frac{1}{|A|}\sum_i \partial_i q_0(A^{-1}(\theta - b))\left(X_i A_i^{-1}(\theta_i - b_i) + y_i\right) \\ &= -\left(\sum_i X_i\right)\pi(g)q_0(\theta) - \sum_i X_i \pi(g)\delta_i q_0(\theta) - \sum_i y_i \pi(g)\partial_i q_0(\theta). \end{aligned}$$

for $g = (A, b) \in G$. Recall $\delta_i q = \theta_i \partial_i q$. Choosing the standard basis in (X, y) we get the basis with $2P$ elements in the statement of the lemma. ☐

With respect to this basis calculating the Fisher matrix consists of calculating $\omega_{\text{Fisher}}(h_i^\bullet, h_j^\bullet)$.

Lemma 5. *Given the basis in Lemma 4, the Fisher information matrix is a block diagonal matrix, with 2×2 symmetric blocks of the form $\left(\begin{smallmatrix} A & B \\ B & C\end{smallmatrix}\right)$ corresponding to pairs of basis elements $\{h_i^X, h_i^y\}$ for $i = 1, \ldots, P$.*

In general $B = \int_{\mathbb{R}}\left(1 + \theta\frac{\tilde{q}_0'(\theta)}{\tilde{q}_0(\theta)}\right)\tilde{q}_0'(\theta)\,\mathrm{d}\theta$. If $\tilde{q}_0$ is symmetric around the origin as the integrand is an odd function $B = 0$. Other entries are given as

$$A = \int_{\mathbb{R}}\left(1 + \theta\frac{\tilde{q}_0'(\theta)}{\tilde{q}_0(\theta)}\right)^2 \tilde{q}_0(\theta)\,\mathrm{d}\theta \qquad\qquad C = \int_{\mathbb{R}}\frac{\tilde{q}_0'(\theta)^2}{q_0(\theta)}\,\mathrm{d}\theta.$$

Proof. We first show that for $i \neq j$ the vectors are orthogonal with respect to this metric,

$$\begin{aligned} \omega_{\text{Fisher}}(h_i^X, h_j^X) &= \int_{\mathbb{R}^P}\frac{h_i^X(\theta)}{q_g(\theta)}\frac{h_j^X(\theta)}{q_g(\theta)}q_g(\theta) = \int_{\mathbb{R}^P}\frac{\pi(g)(q_0 + \delta_i q_0)(\theta)}{\pi(g)q_0(\theta)}\frac{\pi(g)(q_0 + \delta_j q_0)(\theta)}{\pi(g)q_0(\theta)}\pi(g)q_0(\theta)\,\mathrm{d}\theta \\ &= \int_{\mathbb{R}^P}\frac{(q_0 + \delta_i q_0)(\theta)}{q_0(\theta)}\frac{(q_0 + \delta_j q_0)(\theta)}{q_0(\theta)}q_0(\theta)\,\mathrm{d}\theta \\ &= \int_{\mathbb{R}}\left(1 + \frac{\theta_i\tilde{q}_0'(\theta_i)}{\tilde{q}_0(\theta_i)}\right)\tilde{q}_0(\theta_i)\,\mathrm{d}\theta_i \int_{\mathbb{R}}\left(1 + \frac{\theta_j\tilde{q}_0'(\theta_j)}{\tilde{q}_0(\theta_j)}\right)\tilde{q}_0(\theta_j)\,\mathrm{d}\theta_j \prod_{k\neq i,j}\int_{\mathbb{R}}\tilde{q}_0(\theta_k)\,\mathrm{d}\theta_k. \end{aligned}$$

The θ_k integrals are 1, and the other two integrals both vanish as

$$\int_{\mathbb{R}}\left(1 + \theta\frac{\tilde{q}_0'(\theta)}{\tilde{q}_0(\theta)}\right)\tilde{q}_0(\theta)\,\mathrm{d}\theta = 1 + \int_{\mathbb{R}}\theta\tilde{q}_0'(\theta)\,\mathrm{d}\theta$$

integration by parts on the last integral gives -1 hence the integral vanishes.

It should be clear from this calculation that also $\omega_{\text{Fisher}}(h_i^X, h_j^y) = \omega_{\text{Fisher}}(h_i^y, h_j^y) = 0$ for $i \neq j$, and that $\omega_{\text{Fisher}}(h_i^X, h_i^X) = A$, $\omega_{\text{Fisher}}(h_i^X, h_i^y) = B$ and $\omega_{\text{Fisher}}(h_i^y, h_j^y) = C$ for all i. ☐

Let us now write some special cases of distributions.

For the normal distribution $q_0(\theta) = \frac{1}{\sqrt{2\pi}}e^{-\theta^2/2}$ we have that these blocks are of the form $\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1\end{smallmatrix}\right)$. For the Cauchy distribution $\tilde{q}_0(\theta) = \frac{1}{\pi}\frac{1}{1+\theta^2}$ we have the Fisher block $\begin{pmatrix} 1/2 & 0 \\ 0 & 1/2\end{pmatrix}$.

A.5.3 The Differential of $\mathcal{E}$

As per (22) and (23), we need to make several calculations in this specific case

$$\frac{\mathrm{d}}{\mathrm{d}t}(e^{t(X,y)} \cdot \theta)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}\left(e^{tX}\theta + \frac{e^{tX} - \mathbf{1}}{tX}ty\right)\Big|_{t=0} = X\theta + y \tag{27}$$

and similarly for $g = (A, b)$

$$\mathrm{Ad}_{(A,b)}(X, y) = \frac{\mathrm{d}}{\mathrm{d}t}\left((A, b)e^{t(X,y)}(A^{-1}, -A^{-1}b)\right)\Big|_{t=0} = (X, Ay - Xb). \tag{28}$$

For the second calculation note that

$$(A, b)\left(e^{tX}, \frac{e^{tX} - I}{tX}ty\right)(A^{-1}, -A^{-1}b) = \left(e^{tX}, A\frac{e^{tX} - I}{X}y + b - e^{tX}b\right)$$

making use of the fact that the matrices A and e^{tX} are all diagonal and hence commute. Taking the derivative at $t = 0$ we get the result. Making use of (27) we get that

$$\mathrm{Ad}_{(A,b)}(X, y) \cdot \theta = X(\theta - b) + Ay \tag{29}$$

Lemma 6. *Given* $h_{X,y} = \sum_i X_i h_i^X + y_i h_i^y$ *a tangent vector in* $T_{q_g}\mathcal{Q}$ *the differential is given as* $(\mathrm{d}\mathcal{E}(q_g))[h_{X,y}] = \mathrm{tr}(N_X^\top X) + n_y^\top y$ *where*

$$N_X = \frac{1}{\tau}\,\mathrm{diag}(\mathbb{E}_{q_g}[(\theta - b)\nabla\ell]) - I_{P\times P} \qquad \textit{and} \qquad n_y = \frac{1}{\tau}A\mathbb{E}_{q_g}[\nabla\ell].$$

Note that even though we are using the matrix notation for N_X and X, together with the Frobenius norm, since both are diagonal matrices, this is simply a dot product of their diagonal vectors.

Proof. We continue from (22), (23). Due to (27) and (29) we have that

$$\begin{aligned} \textcircled{1} &= \int q_g(\theta)(\nabla\ell(\theta))^\top (X(\theta - b) + Ay)\,\mathrm{d}\theta \\ &= \sum_i X_i \int q_g(\theta)(\theta_i - b_i)\partial_i\ell(\theta)\,\mathrm{d}\theta + \sum_i y_i A_i \int q_g(\theta)\partial_i\ell(\theta)\,\mathrm{d}\theta \end{aligned}$$

and

$$\begin{aligned} \textcircled{2} &= \int (\nabla q_0(\theta))^\top (X\theta + y)\,\mathrm{d}\theta \\ &= \sum_i X_i \int \theta_i\partial_i q_0(\theta)\,\mathrm{d}\theta + \sum_i y_i \int \partial_i q_0(\theta)\,\mathrm{d}\theta \\ &= \sum_i X_i \overset{-1}{\int_{-\infty}^{\infty} \theta_i\tilde{q}_0'(\theta_i)\,\mathrm{d}\theta_i} + \sum_i y_i \overset{0}{\int_{-\infty}^{\infty} \tilde{q}_0'(\theta_i)\,\mathrm{d}\theta_i}. \end{aligned}$$

We get the differentials above when we combine these two calculations. □

A.5.4 The Update Rule

We combine the calculations above assuming $\tilde{q}_0$ is an even function and hence the Fisher matrix is simply diagonal, we write it as $\mathrm{diag}(c_X, c_y)$. This means that

$$(A^{\text{updated}}, b^{\text{updated}}) = (A, b)\exp\left(-\alpha(\tfrac{1}{c_X}n_x, \tfrac{1}{c_y}n_y)\right)$$

Now let $\alpha \mapsto \alpha\tau c_X$ and in separate coordinates this reads as

$$A_i^{\text{updated}} = A_i \exp\left(-\alpha(\mathbb{E}_{q_g}[(\theta_i - b_i)\partial_i\ell] - \tau)\right) \tag{30}$$

$$b_i^{\text{updated}} = b_i + \frac{c_X}{c_y} A_i^2 \frac{\exp\left(-\alpha(\mathbb{E}_{q_g}[(\theta_i - b_i)\partial_i\ell] - \tau)\right) - 1}{\mathbb{E}_{q_g}[(\theta_i - b_i)\partial_i\ell] - \tau} \mathbb{E}_{q_g}[\partial_i\ell]. \tag{31}$$

Notice that the condition $A_i > 0$ is preserved by this update rule. In BLR the positive definiteness of the covariance matrix parameter of a multivariable gaussian distribution cannot be preserved with linear updates, save for very small step sizes. This issue was later remedied by another method by Lin et al. (2020), where the authors used a quadratic approximation to the geodesic on the manifold $\mathcal{Q}$ which also satisfied the positive definiteness constraint.

As a special case consider when q_0 is the Dirac delta distribution at $0 \in \mathbb{R}^P$. Our derivation does not work, but the above update rules are still valid.

The expectation $E_{q_g}[(\theta - b)\nabla\ell] = 0$ in the Dirac delta case, which makes the updates on A components a moot point. This is to be expected since reducing to the Dirac-delta case means willfully forgoing any variance consideration. As for the $b_i = \theta_i$ update the ratio of exponentials only scales our step size and we have

$$\theta^{\text{updated}} = \theta - \alpha'\nabla\ell(\theta)$$

where $\alpha' = \alpha\frac{c_X}{c_y}\frac{e^{\alpha T}-1}{\alpha T}$ is the modified step size. In other words we simply get the usual gradient descent update rule.

A.6 Linear approximation gives the BLR

A.6.1 The multiplicative case

The Bayesian Learning Rule (BLR) of Khan and Rue (2021) is given for exponential families, but our scheme also includes exponential families such as the family of exponential distributions $\tilde{q}_0 = e^{-\theta}$ as mentioned in the beginning of this section. Then $q_{1/\lambda}(\theta) = (\prod_i \lambda_i)e^{-\langle\theta,\lambda\rangle}$ where $1/\lambda$ is interpreted componentwise.

We write the rule for λ's noting that $g_i^{-1} = \lambda_i$. Then using the linearization of exponential

$$\lambda_i^{\text{updated}} = \lambda_i \exp\left(\alpha(\mathbb{E}_{q_{1/\lambda}}[\theta_i\partial_i\ell(\theta)] - 1)\right) \approx \lambda_i(1-\alpha) + \alpha\lambda_i\mathbb{E}_{q_{1/\lambda}}[\delta_i\ell] \tag{32}$$

where we took the linear approximation of the exponential. Note that the Fisher matrix $F_{q_{1/\lambda}} = \operatorname{diag}(\lambda_i^{-2})$ and that

$$\nabla_\lambda\mathbb{E}_{q_{1/\lambda}}[\ell] = \operatorname{diag}\left(\tfrac{1}{\lambda_i}\mathbb{E}_{q_{1/\lambda}}[\delta_i\ell]\right).$$

The right hand side of (32) is exactly the rule of Khan and Rue (2021) applied to the family of exponential distributions.

A.6.2 The affine case

Again we expect that the linear approximation to the above update rule to give us the update rule from Khan and Rue (2021) when we are in the case of the diagonal Gaussian distributions. This happens under temperature $\tau = 1$.

If $q_0(\theta) = \frac{1}{\sqrt{2\pi}^P}e^{-\frac{1}{2}\theta^\top I\theta}$ then we see that the family is the space of diagonal-covariance Gaussian distributions.

$$q_{(A,b)}(\theta) = \frac{1}{\sqrt{2\pi}^P|A|}e^{-\frac{1}{2}(\theta-b)^\top(A^2)^{-1}(\theta-b)}$$

therefore we see that in the notation of Normal distributions $\mathcal{N}(m,\Sigma)$ we have $A^2 = \Sigma = S^{-1}$ (all diagonal matrices in our case) and $b = m$.

For b the update rule has a linear approximation in α,

$$b^{\text{updated}} \approx b - 2\alpha A^2\mathbb{E}_{q_g}[\nabla\ell]$$

where we used that $\frac{e^{-\alpha X}-1}{X} = -\alpha + \alpha^2 X/2 + \cdots$ and that for our chosen $\tilde{q}_0$ that $c_X/c_y = 2$.

This agrees with the first part of Khan and Rue (2021, Eq. 12) save for the factor of 2.

The second part of the same equation is written in terms of an S update, so we take (30) and turn it into an update about S by inverting and squaring both sides: $S_i^{\text{updated}} = S_i \exp\big(2\alpha(\mathbb{E}_{q_g}[(\theta_i - b_i)\partial_i \ell] - 1)\big)$

Taking the linear approximation in α,

$$S_i^{\text{updated}} \approx S_i(1 - 2\alpha) + 2\alpha \mathbb{E}_{q_g}[(\theta_i - b_i)\partial_i \ell]$$

where S may be considered as either a vector or a diagonal matrix. The right hand side may also be written as

$$S^{\text{updated}} \approx S(1 - 2\alpha) + 2\alpha \mathbb{E}_{q_g}[\nabla^2 \ell]$$

following an application of integration by parts using the special form of the Gaussian measure q_g. This is exactly the second update rule in Khan and Rue (2021, Eq. 12) (with $\alpha \mapsto \alpha/2$).

B DETAILS OF THE EXPERIMENTS

For MNIST, no data augmentation was considered. All CIFAR and TinyImageNet experiments use basic data augmentations (random horizontal flipping and cropping). To account for the data augmentation, we set $N \leftarrow 4 \cdot N$. All hyper parameters were selected via a grid search over a moderate amount of configurations and we selected the ones giving the best results for each method.

B.1 Additive vs. multiplicative learning

In all experiments, the learning rate is annealed to zero using a cosine scheduler, and we used 5 "warm-up" epochs where α is linearly increased from zero to the starting learning rate.

MLP. We used a fully connected network with 5 hidden layers (1024, 512, 256, 256, 256 neurons) and $\tanh$ nonlinearity. The regularizer was fixed to $R(\theta) = 125 \cdot \|\theta\|^2$. We train for 60 epochs. The additive update (Alg. 1) runs with $\alpha = 0.01$, $\beta = 0.9$, $q_0 = \mathcal{N}(0, 0.001)$, For the multiplicative update (Alg. 2) we used $\alpha = 50$, $\beta = 0.9$, set q_0 as a Rayleigh distribution (with parameter set to 1), and fixed $\tau = 0.005$. Both methods use $K = 32$ MC samples and a batch size of $n = 50$.

CNN. The convolutional neural network is a basic LeNet-5 architecture with 128, 256 and 512 convolutional filters in the layers. Each convolution is followed by a max-pooling and we used two fully connected layers with 512 and 256 neurons at the end of the network. The regularizer was fixed to $R(\theta) = \|\theta\|^2$. We train for 180 epochs. The additive and multiplicative updates were run using the same parameters as described in the MLP paragraph, except that we used $\alpha = 100$ and $\tau = 0.001$ for the multiplicative updates. For the CNN experiments, both methods use $K = 10$ MC samples and batch size $n = 100$.

Algorithm	α	β_1	β_2	damping
Affine rule (Alg. 3)	1	0.8	0.999	–
Additive rule (Alg. 1)	0.1	0.8	–	–
iVON (Lin et al., 2020, Figure 1)	0.5	0.8	0.999	1
VOGN (Osawa et al., 2019)	0.002	0.95	0.999	0.01
SGD	0.1	0.8	–	–

Table 3: Hyperparameters for the experiments shown in Table 2.

B.2 The affine learning rule

In all experiments, we use the ResNet-20 architecture (as in He et al. (2016, Table 6)) with filter response normalization and train for 180 epochs. The learning rate is decayed to zero using a cosine learning rate scheduler. All methods use one MC sample ($K = 1$). The regularizer is fixed to $R(\theta) = 50 \cdot \|\theta\|^2$.

The hyperparameters for the individual methods are fixed across the data sets and given in Table 3.

The three different choices of q_0 are: $\theta \sim \mathcal{N}(0, 25 \cdot 10^{-6})$, $\theta \sim 0.005 \cdot \text{Laplace}(0, 1)$, and $\theta \sim \text{Uniform}[-0.0025, 0.0025]$.