SELF-CERTIFYING CLASSIFICATION BY LINEARIZED DEEP ASSIGNMENT

BASTIAN BOLL, ALEXANDER ZEILMANN, STEFANIA PETRA, CHRISTOPH SCHNÖRR

Abstract. We propose a novel class of deep stochastic predictors for classifying metric data on graphs within the PAC-Bayes risk certification paradigm. Classifiers are realized as linearly parametrized deep assignment flows with random initial conditions. Building on the recent PAC-Bayes literature and data-dependent priors, this approach enables (i) to use risk bounds as training objectives for learning posterior distributions on the hypothesis space and (ii) to compute tight out-of-sample risk certificates of randomized classifiers more efficiently than related work. Comparison with empirical test set errors illustrates the performance and practicality of this self-certifying classification method.

Contents

1. Introduction

1.1. **Overview, Related Work.** Self-certified learning is the task of using the entirety of available data to find a good model and to simultaneously certify its performance on unseen data from the same underlying distribution. This is opposed to the classic two-stage paradigm in machine learning which first finds a model by using part of the data and subsequently estimates its generalization on held-out test data.

Because the true distribution of data is typically unknown, self-certified learning relies on upper-bounding model *risk* through statistical learning theory. Recently, the PAC-Bayes (Probably Approximately Correct) paradigm [Cat07, Gue19] has attracted much attention due to the recent demonstration of tight risk bounds for deep stochastic neural networks in [DR18]. The authors exploit a PAC-Bayes risk bound by, firstly, training a prior through empirical risk minimization and, secondly, by training a Gibbs posterior distribution. Building on this, recent work [PORSTS21] evaluated various *relaxed PAC-Bayes-kl inequalities*[1] [LS01], including a new one. They showed that non-vacuous risk certificates can be determined numerically which are informative of the out-of-sample error, and that using relaxed upper bounds of the risk for training enables the use of the whole data set for both learning a predictor and certifying its risk.

Similar to [PORSTS21], our approach is to find a PAC-Bayes posterior distribution by optimizing the PAC-Bayes-λ inequality introduced by [TIWS17]. This relaxed PAC-Bayes-kl inequality was shown to be quasiconvex in the parameter that trades off empirical error against model complexity in terms of KL divergence, which is convenient for optimization.

A key component of PAC-Bayes bounds is the empirical risk of stochastic classifiers. In the context of deep learning, such classifiers may be obtained by randomizing neural network weights which typically leads to analytically intractable empirical risk. [LC02] therefore suggest using an upper bound via Monte-Carlo sampling, which holds with high probability and still achieves PAC risk certification with modified probability of correctness. In order to train stochastic classifiers by optimizing PAC inequalities with differentiable surrogate loss, the gradient of empirical risk can similarly be estimated stochastically. [PORSTS21] choose the pathwise gradient estimator [Pri58] and call the resulting framework PAC-Bayes with Backprop, reminiscent of the Bayes-by-Backprop paradigm [BCKW15].

Here, we propose a way to achieve computational tractability of empirical risk in PAC-Bayes without the need for stochastic estimators. Key is the construction of a specific hypothesis class which separates stochasticity from feature extraction by building on certain geometric neural ODEs called *assignment flows* [ÅPSS17]. After suitable parameterization and linearization, the uncertainty quantification approach proposed in [GAZS22] allows to push forward intrinsic normal distributions of initial assignment states in closed

[1]The lowercase kl refers to the relative entropy of two Bernoulli distributions.

form, which can be leveraged to build deep stochastic classifiers with tractable empirical risk.

1.2. **Contribution.** We adopt the PAC-Bayes-λ inequality [TIWS17] to work out a two-stage method as in [DR18, PORSTS21] for evaluating relaxed PAC-Bayes-kl bounds and achieve favorable computational properties compared to prior work.

To this end, we propose a generalized, deep classification variant of S-assignment flows [SS21] and compute the corresponding pushforward distribution in closed form, building on [GAZS22]. Separating stochasticity from feature extraction, we use the computationally tractable pushforward distribution to transform the empirical risk of stochastic classifiers. This requires to compute the pushforward *only once* and subsequently perform very cheap sampling of a transformed integrand in Monte-Carlo methods. Finally, we show that for not too large numbers $c \approx 10$ of classes, much more efficient deterministic Quasi-Monte-Carlo integration [DKS13] can replace Monte-Carlo estimation when evaluating risk certificates. We also show that this deterministic method commutes with backpropagation, i.e., the true gradient is well approximated by backpropagation.

Altogether, this enables us to evaluate the empirical risk and its gradient very efficiently, and to optimize the risk bound provided by the PAC-Bayes-λ inequality with respect to both the posterior and the trade-off between empirical loss and deviation from a data-dependent prior distribution. As a consequence, the *linearized deep assignment flow approach* to classification (Figure 1) becomes a *self-certifying learning method*. We verify its performance and the tightness of risk certificates by a comparison to the empirical test error.

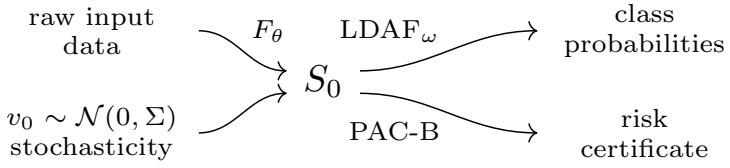

FIGURE 1. Proposed architecture for self-certifying stochastic classifiers. Features F_θ are extracted from input data and define the initial state $S_0 \in \mathcal{W}$ of assignment dynamics. The initialization is randomized on the tangent space $\mathcal{T}_0$ leading to a stochastic classifier LDAF$_\omega$ after forward integration. A PAC-Bayes bound (PAC-B) is employed for computing non-vacuous certificates (upper bound) of the classifier's risk.

2. Background

2.1. **(S-)Assignment Flows.** The *assignment flow approach* [ÅPSS17, Sch20] denotes a class of dynamical systems for analyzing metric data on a graph $G = (V, E)$, $|V| = n$, that is derived in a straightforward way: represent local decisions as point ('state') on a task-specific statistical manifold and perform *contextual structured* decisions by the interaction of these states over the underlying graph. As for the *classification* task considered here, the statistical manifold is the probability simplex equipped with the Fisher-Rao metric of information geometry [AJVLS17], and the interaction corresponds to *geometric* state averaging derived from the affine e-connection. Adopting the parametrization of [SS21], the resulting dynamical system reads

$$\dot{S}(t) = R_{S(t)}[\Omega S(t)], \qquad S(0) = S_0, \tag{1}$$

where $S(t) \in \mathcal{W} \subset \mathbb{R}^{n \times c}_{++}$ comprises the state at each vertex $i \in V$ as row vector $S_i(t) \in \mathbb{R}^c_{++}$, where $2 \leq c \in \mathbb{N}$ denotes the number of classes. The underlying geometry always restricts $S(t)$ to the set of row-stochastic matrices with full support called *assignment manifold* $\mathcal{W}$, with trivial tangent bundle $T\mathcal{W} = \mathcal{W} \times \mathcal{T}_0$, where

$$\mathcal{T}_0 = \{V \in \mathbb{R}^{n \times c} \colon \langle \mathbb{1}_c, V_i \rangle = 0,\ i \in V\}. \tag{2}$$

The right-hand side of (1) constitutes a proper vector field on $\mathcal{W}$ that is parametrized by the matrix Ω and mapped to $T\mathcal{W}$ by the state-dependent non-orthogonal projection mapping that acts row-wise by

$$R_S[\Omega S] = \big(R_{S_1}(\Omega S)_1, \ldots, R_{S_n}(\Omega S)_n\big)^\top, \tag{3}$$

where $S_i, (\Omega S)_i$ denote the ith row vectors and

$$R_{S_i} = \operatorname{Diag}(S_i) - S_i S_i^\top \in \mathbb{R}^{c \times c}, \quad i \in V. \tag{4}$$

Each row of (1) defines a *replicator equation* [HS03] that are *coupled* over the graph through the right-hand side and interact by geometric numerical integration [ZSPS20] of the assignment flow $S(t)$. Under mild conditions on Ω, $\lim_{t \to \infty} S(t)$ converges to hard label assignments and is *stable* against data perturbations [ZZS21].

As a consequence, (1) may be seen as a particular system of neural ODEs [CRBD18] that represent the layers of a deep network by time-discrete geometric numerical integration of the flow. In Section 3, we adopt a 'deep structured' parametrization of (1) and restrict ourselves to a *linearization* of the resulting large-scale dynamical system.

Our main contribution is to show that a state-of-the-art PAC Bayes bound [TIWS17] can be evaluated both rigorously and efficiently for the corresponding hypothesis class of *linearized deep assignment flows* (Section 4) and lead to *non-vacuous quantitative* risk certificates (Section 5).

2.2. **PAC-Bayes Risk Certification.** Consider stochastic classifiers, i.e., distributions μ over a hypothesis space $\mathcal{H}$, elements of which are functions $\phi_\theta \colon \mathcal{D} \to \mathcal{T}_0$. Suppose $\mathcal{H}$ is parameterized by $\theta \in \Theta$ and identify distributions ρ over $\mathcal{H}$ with distributions over the parameter space Θ. For given loss function $\ell \colon \mathcal{T}_0 \to \mathbb{R}$ and a generally unknown data distribution $\mathfrak{D}$ over $\mathcal{D} \times \mathcal{T}_0$, the goal of learning stochastic classifiers is to find μ such that the expected risk

$$\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}(\theta)] := \mathbb{E}_{\theta\sim\mu}\big[\mathbb{E}_{(x,y)\sim\mathfrak{D}}\ell(\phi_\theta(x), y)\big] \tag{5}$$

is minimized. Since $\mathfrak{D}$ is unknown, the true risk $\mathfrak{L}(\theta)$ is difficult to estimate. A tractable related quantity is the *empirical risk*

$$\mathfrak{L}_m(\theta) := \frac{1}{m}\sum_{k\in[m]} \ell(\phi_\theta(x_k), y_k) \tag{6}$$

where (x_k, y_k) denote m i.i.d. samples drawn from $\mathfrak{D}$. PAC-Bayesian theory [Cat07, Gue19] considers a distribution μ called *PAC-Bayes posterior* which depends on the sample as well as a reference distribution π called *PAC-Bayes prior* which does not depend on the *same* sample. A goal is to construct tight, high-confidence bounds on (5) which only depend on tractable quantities such as (6). In our analysis, we use the following state-of-the-art bound.

Theorem 2.1 (**PAC-Bayes-λ Inequality** [TIWS17])**.** *For any $\epsilon > 0$ and any $\lambda \in (0, 2)$, it holds with probability at least $1 - \epsilon$ for all posterior distributions μ over parameters θ simultaneously*

$$\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}(\theta)] \leq \frac{\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}_m(\theta)]}{1 - \frac{\lambda}{2}} + \frac{\mathrm{KL}(\mu \colon \pi) + \log\frac{2\sqrt{m}}{\epsilon}}{m\lambda(1 - \frac{\lambda}{2})}, \tag{7}$$

where m denotes the size of an i.i.d. sample set.

Regarding the evaluation of the right-hand side, key issues are the definition of prior and posterior distributions π, μ over the hypothesis space and the accurate and efficient computation of the *expected* empirical risk $\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}_m(\theta)]$, which typically is a hard task in practice. We deal with these issues in Sections 4.1, 4.2 and 4.3, 4.4, respectively.

3. Deep Assignment Flows

3.1. **Deep S-Flows.** Motivated by the use of coupled replicator dynamics in game theory [MM15], we generalize S-flows (1) by enabling additional interaction on the label space. To shorten notation, vectorize S, V, R_S given by (1)–(4) and the orthogonal projection $\Pi_0 \colon \mathbb{R}^{|V|\times c} \to \mathcal{T}_0$ according to

$$s := \mathrm{vec}(S), \qquad v := \mathrm{vec}(V) \tag{8}$$

$$\Pi_0^{\mathfrak{v}} v := \mathrm{vec}(\Pi_0 V), \qquad R_s^{\mathfrak{v}} v := \mathrm{vec}(R_S V) \tag{9}$$

$$\exp_s^{\mathfrak{v}} v := \mathrm{vec}(\exp_S V), \tag{10}$$

where $\exp_S(V) = (\exp_{S_1}(V_1), \ldots, \exp_{S_1}(V_1)) \in \mathcal{W}$ and $\exp_{S_i}(V_i) = \frac{S_i e^{V_i}}{\langle S_i, e^{V_i}\rangle}$ with componentwise multiplication in the numerator. In vectorized notation, the S-flow dynamics (1) read

$$\dot{s}(t) = R^{\mathfrak{v}}_{s(t)}(\Omega \otimes \mathbb{I}_c)s(t), \qquad s(0) = s_0 = \mathrm{vec}(S_0)\ . \tag{11}$$

We now generalize by breaking up the Kronecker product structure of $\Omega \otimes \mathbb{I}_c$. Re-using the symbol Ω to denote a matrix

$$\Omega \in \mathbb{R}^{N\times N}, \qquad N = cn \tag{12}$$

we define the *deep assignment flow* (DAF) in vectorized form as

$$\dot{s}(t) = R^{\mathfrak{v}}_{s(t)}\Omega s(t), \qquad s(0) = s_0\ . \tag{13}$$

This class of dynamics is more general than (1) while remaining amenable to lifting and linearization with minimal modifications to other assignment flows [ZSPS20, BSS21]. Concerning the PAC-Bayes risk certification, we observe that (13) typically leads to better generalization and more gain between posterior and prior as compared to (1).

3.2. **Classification by Deep Assignment.** Unlike typical assignment flow approaches, our aim is not to perform image labeling (i.e., segmentation) but classification. To this end, we choose the underlying graph to be relatively small ($n = 50$ nodes) and densely connected with learned symmetric matrix $\Omega \in \mathbb{R}^{N\times N}$ (cf. (12)). We also designate a single node to carry class probabilities. Through the dynamics (13), the state of this node will evolve towards an integer assignment, i.e., a class decision. By convention, let the classification node be the node with index 1 and let

$$\mathcal{I} = [c] := \{1, \ldots, c\} \tag{14}$$

denote the set of indices such that $s_{\mathcal{I}} = S_1$ contains classification probabilities. Further, denote the relative interior of a single probability simplex with c corners by $\mathcal{S}_c$.

3.3. **Linearizing Deep Assignment Flows.** We parameterize DAFs in the tangent space of $\mathcal{W}$ at S_0 by $s(t) = \exp^{\mathfrak{v}}_{s_0}(v(t))$ where $v(t)$ follows

$$\dot{v}(t) = \Pi^{\mathfrak{v}}_0 \Omega \exp^{\mathfrak{v}}_{s_0}(v(t)), \qquad v(0) = v_0 = 0 \tag{15}$$

and compute using $d\exp_{S_{0;i}}(V_{0;i})[U] = R_{S_{0;i}}[U]$

$$\Pi^{\mathfrak{v}}_0 \Omega \exp^{\mathfrak{v}}_{s_0}(v_0) = \Pi^{\mathfrak{v}}_0 \Omega s_0 \tag{16}$$

$$\mathrm{d}(\Pi^{\mathfrak{v}}_0 \circ \Omega \circ \exp^{\mathfrak{v}}_{s_0})(v_0)[u] = \Pi^{\mathfrak{v}}_0 \Omega \mathrm{d}(\exp^{\mathfrak{v}}_{s_0})(v_0)[u] = \Pi^{\mathfrak{v}}_0 \Omega R^{\mathfrak{v}}_{s_0} u \tag{17}$$

which yields the *linearized* DAF

$$\dot{v}(t) = \underbrace{\Pi^{\mathfrak{v}}_0 \Omega R^{\mathfrak{v}}_{s_0}}_{=:A} v(t) + \underbrace{\Pi^{\mathfrak{v}}_0 \Omega s_0}_{=:v_D} \tag{18}$$

The solution in closed form reads

$$v(t) = t\varphi(tA)v_D = t\varphi\left(t\Pi^{\mathfrak{v}}_0 \Omega R^{\mathfrak{v}}_{s_0}\right)\Pi^{\mathfrak{v}}_0 \Omega s_0 \tag{19}$$

and Krylov methods for evaluating the analytical matrix-valued function $\varphi(z) = \frac{e^z-1}{z}$ as well as respective gradient approximations computed in [ZPS21] apply without modification. Note that even though $\varphi(tA)$ acts linearly on v_D in (19), LDAF dynamics are much more capable than a learned linear map $\mathcal{T}_0 \to T_0\mathcal{S}_c$. This is due to the fact that $A = \Pi_0^{\mathfrak{v}}\Omega R_{s_0}^{\mathfrak{v}}$ depends on s_0 so *each* input datum is transformed by a *different* linear operator.

4. Risk Certification of Stochastic LDAF Classifiers

We consider PAC-Bayes risk certificates which bound the expected risk of a stochastic classifier (see Section 2.2). The evaluation of such a certificate requires evaluation of expected empirical risk which presents a computational challenge. To mitigate this, one may use Monte-Carlo methods to upper-bound the expected empirical risk with high probability as proposed in [LC02]. Here, we propose instead a strategic choice of hypothesis class and shape of stochastic classifiers which allows to directly compute the expected empirical risk efficiently and precisely while also allowing for the use of deep feature extractors.

4.1. LDAF Hypothesis Space.

We define the hypothesis space $\mathcal{H}$ of classifiers ϕ built by composing a feature extractor with LDAF dynamics (18) up to time $T > 0$. To this end, fix a small, densely connected graph. We use $n = 50$, $c = 10$ in our experiments. This defines an associated assignment manifold $\mathcal{W}$ and tangent space $\mathcal{T}_0$ on which a linear operator Ω specifies DAF dynamics (15) with linearization (18). We assume Ω is symmetric and denote the vector of learnable parameters defining Ω by ω. For a given data point x in some vector space $\mathcal{D}$ such as the space of RGB images, a corresponding initial point $s_0 \in \mathcal{W}$ is computed by extracting features using a neural network $F_\vartheta \colon \mathcal{D} \to \mathcal{T}_0$ with parameters ϑ and setting $s_0 = \exp_{\mathbb{1}_\mathcal{W}}(F_\vartheta(x))$. Following linearization of the DAF vector field, we take the initialization $v_0 \in \mathcal{T}_0$ as additional parameters. Forward integration up to time T gives a state $s(T) = \exp_{s_0}(v(T)) \in \mathcal{W}$ which contains class probabilities $S(T)_1 \in \mathcal{S}_c$ at the classification node. We collect the described sequence of operations on $\mathcal{W}$ into a function ψ_{ω,v_0} and call

$$\mathcal{H} = \{\phi \colon \mathbb{R}^d \to \mathcal{S}_c \mid \phi = \psi_{\omega,v_0} \circ \exp_{\mathbb{1}_\mathcal{W}} \circ F_\vartheta\} \tag{20}$$

the hypothesis class of *LDAF classifiers*. Measures on $\mathcal{H}$ are identified with measures on the parameter space $\overline{\mathcal{H}}$ which contains triples $\theta = (\vartheta, \omega, v_0)$.

Denote by $\mathcal{P}$ the class of probability measures μ on $\mathcal{H}$ with shape

$$\mu = \delta_\vartheta \times \delta_\omega \times \mathcal{N}(0, \Sigma_0) \tag{21}$$

where $\mathcal{N}(0, \Sigma_0)$ denotes an intrinsic normal distribution on $\mathcal{T}_0$ (cf. chapter 3 in [RH05]) centered at 0. Due to the structure of $\mathcal{T}_0$ as a linear subspace of $\mathbb{R}^N$, covariance matrices $\Sigma_0 \in \mathbb{R}^{N\times N}$ are positive semi-definite. Each measure in $\mathcal{P}$ corresponds to a stochastic LDAF classifier which operates by

taking an independent sample from μ for each datum. In order to compute PAC-Bayes risk certificates, we need to compute the expected empirical risk of stochastic classifiers as well as their complexity with respect to a reference distribution. Suppose the reference distribution π (PAC-Bayes prior) also has shape (21). Further, ω and ϑ are fixed and only the distribution of v_0 differs between π and μ (PAC-Bayes posterior). The following lemma asserts that within the constructed setting, model complexity in terms of relative entropy is well-defined and computationally feasible.

Lemma 4.1. *Fix the distributions $\mu = \delta_\vartheta \times \delta_\omega \times \mathcal{N}(0, \Sigma_0)$ and $\pi = \delta_\vartheta \times \delta_\omega \times \mathcal{N}(0, \widetilde{\Sigma}_0)$ in $\mathcal{P}$. Then μ is absolutely continuous with respect to π ($\mu \ll \pi$) and*

$$\mathrm{KL}[\mu \colon \pi] = \mathrm{KL}[\mathcal{N}(0, \Sigma_0) \colon \mathcal{N}(0, \widetilde{\Sigma}_0)] \tag{22}$$

Proof. Let A be a measurable subset of $\overline{\mathcal{H}}$ with $\pi(A) = 0$. Then

$$\delta_\vartheta(A_1)\delta_\omega(A_2)\mathcal{N}(0, \widetilde{\Sigma}_0)(A_3) = 0 \tag{23}$$

for projections A_1, A_2, A_3 of A onto the respective coordinates. Therefore, at least one of the factors needs to vanish. If either of the first two vanishes, this directly implies $\mu(A) = 0$. In addition, $\mathcal{N}(0, \widetilde{\Sigma}_0)(A_3) = 0$ implies $\mathcal{N}(0, \Sigma_0)(A_3) = 0$ and thus $\mu(A) = 0$ because both normal distributions define equivalent measures. It follows $\mu \ll \pi$. Because the three factors in μ resp. π are independent, the relative entropy decomposes as

$$\mathrm{KL}[\mu \colon \pi] = \mathrm{KL}[\mathcal{N}(0, \Sigma_0) \colon \mathcal{N}(0, \widetilde{\Sigma}_0)] + \underbrace{\mathrm{KL}[\delta_\vartheta \colon \delta_\vartheta]}_{=0} + \underbrace{\mathrm{KL}[\delta_\omega \colon \delta_\omega]}_{=0} \tag{24}$$

□

4.2. **Data-Dependent Prior.** Recently, the use of data for finding a good prior π has been identified as critical for obtaining sharp generalization bounds. This development was sparked by non-vacuous risk bounds for neural networks achieved by [DR18]. Unlike this work, we do not make use of differential privacy to account for sharing data between prior and posterior. Instead, we forego potentially more efficient use of data in favor of simplicity by splitting the available dataset into a *training* and a *validation* set. The training set is used to compute a PAC-Bayes prior distribution π via empirical risk minimization. The validation set is subsequently used to fine-tune the PAC-Bayes posterior distribution μ by minimizing a risk bound for a differentiable surrogate loss starting from π. In addition, the validation set is also used to evaluate the final classification risk certificate. In order to achieve the setting assumed in Lemma 4.1, we only vary the distribution of v_0 when optimizing μ and keep the feature extraction parameters ϑ as well as the fitness parameters ω fixed.

4.3. **Computing the Expected Empirical Risk.** We now aim to leverage the analytical tractability of LDAF forward integration to efficiently compute the empirical risk of stochastic classifiers in $\mathcal{P}$. This can be done irrespective of feature extraction because stochasticity only pertains to the LDAF initialization v_0. Key to the construction is the ability to push forward a multivariate normal distribution on $\mathcal{T}_0$ under LDAF dynamics in closed-form. This amounts to an extension of the uncertainty quantification approach [GAZS22] to the deep flows considered here.

Proposition 4.2 (**LDAF Pushforward**). *Consider the LDAF dynamics* (18) *and let* $v(0) \sim \mathcal{N}(0, \Sigma_0)$. *Then* $v(t)$ *follows the multivariate normal distribution* $\eta(t) = \mathcal{N}(\mathfrak{m}(t), \Sigma(t))$ *for every* $t > 0$ *with moments*

$$\mathfrak{m}(t) = t\varphi(tA)b, \tag{25a}$$

$$\Sigma(t) = \operatorname{expm}(tA)\Sigma_0 \operatorname{expm}(tA)^\top \tag{25b}$$

Proof. For $v_0 \neq 0$, the closed form solution (19) is modified to

$$v(t) = \operatorname{expm}(tA)v_0 + t\varphi(tA)b. \tag{26}$$

We see that for fixed $t > 0$, $v(0)$ is mapped to $v(t)$ by an affine transformation. Therefore, $v(t)$ still follows a multivariate normal distribution. Denote its moments by

$$\mathfrak{m}(t) = \mathbb{E}[v(t)], \tag{27a}$$

$$\Sigma(t) = \mathbb{E}[(v(t) - \mathfrak{m}(t))(v(t) - \mathfrak{m}(t))^\top]. \tag{27b}$$

One readily computes

$$\dot{\mathfrak{m}}(t) = \mathbb{E}[\dot{v}(t)] = \mathbb{E}[Av(t) + b] = Am(t) + b \tag{28}$$

which has the closed form solution

$$\mathfrak{m}(t) = t\varphi(tA)b \tag{29}$$

because $\mathfrak{m}(0) = \mathbb{E}[v(0)] = 0$. A straightforward computation shows that

$$\dot{\Sigma}(t) = \mathbb{E}[\dot{v}(t)(v(t) - \mathfrak{m}(t))^\top + (v(t) - \mathfrak{m}(t))\dot{v}(t)^\top] \tag{30}$$

with $\Sigma(0) = \Sigma_0$. Inserting the shape of (18) now gives

$$\dot{\Sigma}(t) = A\Sigma(t) + \Sigma(t)A^\top \tag{31}$$

with closed form solution

$$\Sigma(t) = \operatorname{expm}(tA)\Sigma_0 \operatorname{expm}(tA)^\top \tag{32}$$

analogous to the computation in [GAZS22]. □

The full covariance matrix $\Sigma(t) \in \mathbb{R}^{N\times N}$ in (31) is quite large ($N = nc$ as in (12)) and expensive to compute. However, for the purpose of classification, we only need the marginal $\eta^{(1)}(T)$ of $\eta(T)$ for the classification node (the first node by convention). Because $\eta(T)$ is a normal distribution, the moments of $\eta^{(1)}(T)$ are the subvectors of (25) built from all entries with indices $\mathcal{I}$ (cf. (14)). We may now leverage the available closed form (25) to transform

the empirical risk of stochastic LDAF classifiers, which is the main technical contribution of this paper.

Theorem 4.3 (**LDAF Expected Empirical Risk**). *Fix (linear) coordinates of $T_0\mathcal{S}_c$ by choosing the columns of*

$$P := \begin{pmatrix} \mathbb{I}_{c-1} \\ -\mathbb{1}_{c-1}^\top \end{pmatrix} \in \mathbb{R}^{c\times(c-1)} \tag{33}$$

as basis vectors. For a given data sample $\{(x_k, y_k)\}_{k\in[m]}$ and loss function $\ell\colon T_0\mathcal{S}_c \times [c] \to \mathbb{R}$, the stochastic classifier with distribution $\mu = \delta_\vartheta \times \delta_\Omega \times \mathcal{N}(0, \Sigma_0)$ on the hypothesis class $\mathcal{H}$ has expected empirical risk $\mathbb{E}_{v_0\sim\mu}[\mathfrak{L}_m(v_0)]$ given by

$$\frac{1}{m}\sum_{k\in[m]} \int_{\mathbb{R}^{c-1}} \ell(Pz + F_\vartheta(x_k), y_k)\rho_{\widehat{\mathfrak{m}}(x_k),\widehat{\Sigma}(x_k)}(z)\mathrm{d}z. \tag{34}$$

Here, ρ denotes the density of a multivariate normal distribution with the indicated moments

$$\widehat{\mathfrak{m}}(x_k) = \mathfrak{m}(T)_{\mathcal{I}\setminus\{c\}} \tag{35a}$$

$$\widehat{\Sigma}(x_k) = \Sigma(T)_{\mathcal{I}\setminus\{c\},\mathcal{I}\setminus\{c\}} \tag{35b}$$

which are subvectors resp. submatrices of (25) *for each input datum derived from the marginal distribution $\eta^{(1)}(T)$ of $\eta(T)$ for the classification node. The last index c is omitted due to the shape of basis* (33).

Proof. Note that A and b in (25) depend on the initial assignment state $s_0 = \exp^{\mathfrak{v}}_{\mathbb{1}_{\mathcal{W}}\mathfrak{v}}(F_\vartheta(x))$ according to (18) which clarifies the data dependence of $\widehat{\mathfrak{m}} = \widehat{\mathfrak{m}}(x_k)$. Because $T_0\mathcal{S}_c$ is a vector space, any proper multivariate normal distribution in $T_0\mathcal{S}_c$ corresponds to an improper intrinsic multivariate normal distribution in (linear) coordinates such as in the basis (33). The particular choice of basis (33) is such that mean an covariance are entries of the moments $\widetilde{\mathfrak{m}} = \mathfrak{m}(T)_{\mathcal{I}}$, $\widetilde{\Sigma} = BB^\top$ of $\eta^{(1)}(T)$ because

$$\widetilde{\mathfrak{m}} = \mathbb{E}[v(T)_{\mathcal{I}}] = \begin{pmatrix} \widehat{\mathfrak{m}} \\ -\langle \mathbb{1}_c, \widehat{\mathfrak{m}}\rangle \end{pmatrix} \tag{36a}$$

$$\widetilde{\Sigma} = \mathbb{E}[(v(T)_{\mathcal{I}} - \widetilde{\mathfrak{m}})(v(T)_{\mathcal{I}} - \widetilde{\mathfrak{m}})^\top] \tag{36b}$$

$$= P\mathbb{E}[(v(T)_{\mathcal{I}\setminus\{c\}} - \widehat{\mathfrak{m}})(v(T)_{\mathcal{I}\setminus\{c\}} - \widehat{\mathfrak{m}})^\top]P^\top \tag{36c}$$

$$= \begin{pmatrix} \widehat{\Sigma} & -\widehat{\Sigma}\mathbb{1} \\ -\mathbb{1}^\top\widehat{\Sigma} & -\mathbb{1}^\top\widehat{\Sigma}\mathbb{1} \end{pmatrix}. \tag{36d}$$

We now transform the sought empirical risk by leveraging the shape of pushforward marginal under the LDAF dynamics.

$$\mathbb{E}_{v_0\sim\mu}[\mathfrak{L}_m(v_0)] = \int \frac{1}{m}\sum_{k=1}^{m} \ell(\phi(x_k), y_k)\mathrm{d}\mu(\phi) \tag{37a}$$

$$= \frac{1}{m}\sum_{k=1}^{m} \int_{T_0\mathcal{S}_c} \ell(v + F_\vartheta(x_k)_\mathcal{I}, y_k)\mathrm{d}\eta^{(1)}(v) \tag{37b}$$

$$= \frac{1}{m}\sum_{k=1}^{m} \int_{\mathbb{R}^{c-1}} \ell(Pz + F_\vartheta(x_k)_\mathcal{I}, y_k)\rho_{\widehat{\mathfrak{m}}(x_k),\widehat{\Sigma}(x_k)}\mathrm{d}z \tag{37c}$$

Here, (37b) uses the marginal distribution $\eta^{(1)}$ of the pushforward computed in Proposition 4.2. To obtain class probabilities, the tangent vector resulting from LDAF forward integration needs to be lifted at $s_0 = \exp_{\mathbb{1}_\mathcal{W}}(F_\vartheta(x_k))$. In (37b), we have expressed this by a shift in classification logits due to

$$\exp^{-1}_{\mathbb{1}_\mathcal{W}}(\exp_{s_0}(v)) = \exp^{-1}_{\mathbb{1}_\mathcal{W}}(\exp_{\exp_{\mathbb{1}_\mathcal{W}}(v^0)}(v)) \tag{38a}$$

$$= \exp^{-1}_{\mathbb{1}_\mathcal{W}}(\exp_{\mathbb{1}_\mathcal{W}}(v^0 + v)) \tag{38b}$$

$$= v^0 + v \tag{38c}$$

with $v^0 = \exp^{-1}_{\mathbb{1}_\mathcal{W}}(s_0) = F_\vartheta(x_k)$. □

Let $\sqrt{\cdot}$ denote the matrix square root and define the matrix $B \in \mathbb{R}^{c\times N}$ of the first c rows of $\operatorname{expm}(TA)\sqrt{\Sigma_0}$. Then $BB^\top$ is the covariance matrix of $\eta^{(1)}(T)$ and rows

$$B_i = \sqrt{\Sigma_0}\operatorname{expm}(TA^\top)e_i, \qquad i \in \mathcal{I} = [c] \tag{39}$$

can be efficiently computed by approximating the matrix exponential action via Krylov subspace methods [Saa92, HO10, NW12]. Computing $\eta^{(1)}(T)$ is therefore only roughly c times more expensive than a single LDAF forward pass if the action of $\sqrt{\Sigma_0}$ is efficiently computable. We ensure this by the parametrization

$$\sqrt{\Sigma_0} = (\mathbb{I}_n \otimes P)\big(\operatorname{Diag}(d) + qq^\top\big) \in \mathbb{R}^{N\times N} \tag{40}$$

with P given by (33). This allows to compute the relative entropy (22) using the Sherman-Morrison formula and the matrix determinant lemma such that learning the parameters $d, q \in \mathbb{R}^{(c-1)n}$ by minimizing the PAC-Bayes bound (7) is computationally efficient. Here, we use the parameterization (40) for both PAC-Bayes prior π and posterior μ (cf. Section 5).

4.4. **Numerical Integration.** Previous works on PAC-Bayes risk certification have commonly resorted to approximating the expected empirical risk by a Monte-Carlo (MC) method. This accounts for very high-dimensional domains of integration, but it is computationally expensive because it requires evaluation of the integrand at many sample points which entails a separate forward pass for every drawn sample. Theorem 4.3 proposes a

way to circumvent this problem by computing the pushforward distribution *only once* (at roughly the cost of c forward passes) and by subsequently performing very cheap sampling of the integrand. This amounts to large efficiency gains when using MC methods.

Even though the domain of integration in (37c) is low-dimensional compared to, e.g., the parameter space considered in [PORSTS21] (millions of neural network parameters), standard product cubature rules still are not able to compete with MC as the number of function evaluations for such methods grows exponentially with dimension.

However, Quasi-Monte-Carlo (QMC) methods can improve on MC in the case at hand by leveraging smoothness and moderate dimension. The rationale behind QMC methods is to choose a sequence of deterministic sample points which has lower discrepancy than the uniform random points used in MC. By the Koksma-Hlawka inequality [DKS13][Theorem 3.9], the error of approximate integration via an unweighted sample point mean is bounded by the Hardy-Krause variation of the integrand times the discrepancy of the sample points. Thus, sequences of low-discrepancy sample points, such as the Sobol sequence, asymptotically lead to more efficient integration than MC if the integrand has bounded Hardy-Krause variation. The rate of convergence was shown to further improve under stricter smoothness assumptions [BO16]. In practice, QMC methods are observed to outperform MC particularly for moderate dimension and smooth integrands [MC95]. We observe that relatively few (10K) sample points suffice to compute the empirical risk of stochastic LDAF classifiers with sufficient accuracy. This is not the case of MC as illustrated in Figure 2.

To make QMC methods more easily applicable, we transform the integral in (37c) by successive substitutions

$$\int_{\mathbb{R}^{c-1}} \ell(Pz + F_\vartheta(x_k)_{\mathcal{I}}, y_k) \rho_{\widehat{\mathfrak{m}}(x_k), \widehat{\Sigma}(x_k)} \mathrm{d}z \tag{41a}$$

$$= \int_{[0,1]^{c-1}} \ell(PH\Phi^{-1}(z) + \mathfrak{m}(T) + F_\vartheta(x_k)_{\mathcal{I}}, y_k) \mathrm{d}z \tag{41b}$$

where Φ is elementwise the cumulative distribution function of a standard normal distribution and $\widehat{\Sigma} = HH^\top$ denotes Cholesky decomposition. For moderate c such as the classification problems considered in Section 5 ($c = 10$), QMC integration allows to compute the empirical risk very precisely while using relatively few (10K) sample points.

In addition to gained efficiency, QMC integration has another distinct advantage over MC: it *commutes with backpropagation*. To see this, use the shorthand notation $f(z, \theta)$ for the integrand in (41b) and assume the loss function ℓ is differentiable. QMC integration with deterministic Sobol points $\{z_k\}_{k\in[q]}$ followed by derivation reads

$$\frac{\partial}{\partial \theta_i} \int_{[0,1]^{c-1}} f(z, \theta) \mathrm{d}z \approx \frac{\partial}{\partial \theta_i} \frac{1}{q} \sum_{k\in[q]} f(z_k, \theta) \tag{42}$$

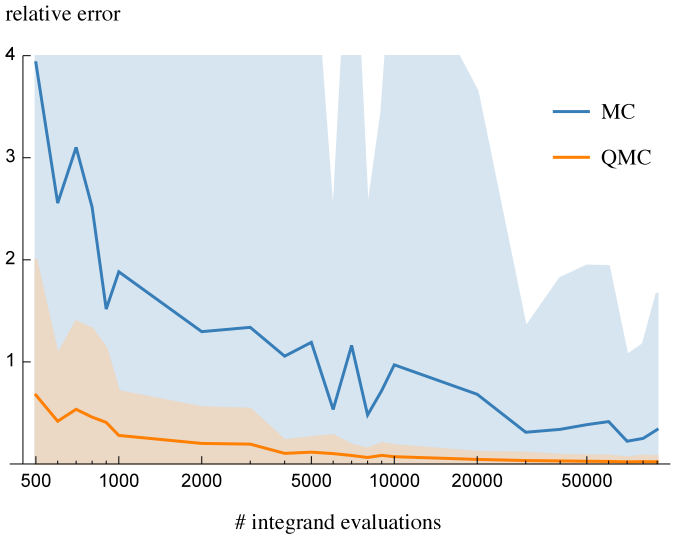

FIGURE 2. Accuracy of QMC integration ■ and MC integration ■ for computing the expected empirical risk (41b) with varying number of sample points. Error bands indicate standard deviation within a batch of 100 CIFAR-10 data points. Because the pushforward distribution of Theorem 4.3 is computationally tractable, sampling is very efficient and computing the reference solution by drawing 100M MC samples only takes minutes on a single GPU. In our proposed QMC method, we compute the pushforward distribution and subsequently perform 10K integrand evaluations at negligible computational cost.

and exchanging summation and differentiation on the r.h.s. yields

$$\frac{1}{q}\sum_{k\in[q]}\frac{\partial}{\partial\theta_i}f(z_k,\theta)\approx\int_{[0,1]^{c-1}}\frac{\partial}{\partial\theta_i}f(z_k,\theta)\mathrm{d}z\ . \tag{43}$$

Thus, if integration and differentiation can be exchanged then backpropagation through QMC integration gives the same result as approximating the exact derivative via QMC.

5. Benchmarks and Discussion

We now compare self-certifying stochastic classifiers trained by minimizing PAC-Bayes generalization bounds to deterministic classifiers evaluated on held-out test data on the CIFAR-10 and FashionMNIST datasets.

5.1. **Training Stochastic LDAF Classifiers.** Consider the image classification task on CIFAR-10 [KH+09] and FashionMNIST [XRV17]. Even though self-certified learning allows to use the entire dataset for training

and judge generalization via risk certification, we still hold out the preassigned test sets for direct comparability with deterministic classifiers. The remaining data are split into a training set used for training priors as well as deterministic classifiers and a validation set ($m = 10\text{k}$) used for training posteriors and evaluating risk certificates.

As a baseline, we use a deterministic ResNet18 classifier [HZRS16] with a slight modification to account for small input size (see Appendix A). We train on the described training split (with held-out validation and test data) for 200 epochs of stochastic gradient descent (weight decay 0.001, momentum 0.9, batch size 128) with a cosine annealing learning rate schedule [LH16] starting at learning rate 0.1 and a light data augmentation regime as in [ZK16].

Stochastic classifiers π and μ are implemented as distributions with shape (21) over the hypothesis space $\mathcal{H}$. The graph G is chosen relatively small ($n = 50$ nodes) and densely connected with symmetric matrix Ω as in (12). For feature extraction F, we replace the classification head of the ResNet18 described above with a dense layer mapping to $\mathcal{T}_0$, i.e., dimension $N = nc = 50 \cdot 10$. Both PAC-Bayes prior π and posterior μ are implemented by randomizing LDAF initialization v_0 on the tangent space $\mathcal{T}_0$ according to a zero-mean multivariate normal distribution with covariance parameterized as in (40).

To train stochastic classifiers, we proceed in two steps. First, using the *same training routine* as for ResNet18, we train a deterministic LDAF classifier on the training split. This defines the mean of stochastic classifiers in $\mathcal{H}$. For the PAC-Bayes prior π, we fix the covariance Σ_0 parameterized according to (40) by drawing the entries of $d, p \in \mathbb{R}^{(c-1)n}$ from a univariate normal distribution centered at 0.1 with variance 0.01. Initializing μ at π, we subsequently train μ by minimizing the r.h.s. of the bound (7), alternating between optimization of μ and λ. According to [TIWS17], the PAC-Bayes-λ bound (7) is strongly quasiconvex as a function of λ under mild conditions, which ensures convergence of alternating optimization. For each alternation, we optimize posteriors over 5 epochs of SGD (learning rate 0.1) on the validation set and find empirically that both λ and μ converge quickly (< 10 alternations). The performances of stochastic classifiers as observed on the held-out test data as well as risk certificates computed on validation data are listed in Table 1. We provide the code used to compute these values as supplementary material[2].

5.2. **Discussion and Conclusion.** As indicated in Table 1, the empirical performance of stochastic classifiers built in the proposed way is close to the performance of a (deterministic) ResNet18 used as a baseline. Notably, this is without altering the training regime.

[2]Code can be found at `https://github.com/IPA-HD/ldaf_classification`

TABLE 1. Empirical performance of deterministic (top) and stochastic (middle) classifiers measured as 01 loss (error percentage) on the held-out test data of CIFAR-10 and FashionMNIST. PAC-Bayes risk certificates (bottom) bound the risk of stochastic classifiers with high probability $1 - \epsilon$. Tightness of risk certificates is on par with the state-of-the-art results (PBB = PAC-Bayes with Backprop) reported in [PORSTS21] and our novel method offers improved computational efficiency due to tractable empirical risk.

	CIFAR-10	FashionMNIST
ResNet18	**5.00**	**4.81**
LDAF Mean	5.28	5.13
Prior (ours)	5.49	5.13
Posterior (ours)	**5.31**	5.12
Posterior PBB	14.75	
Cert. (ours) $\epsilon = 0.01$	**6.36**	6.07
Cert. (ours) $\epsilon = 0.05$	**6.19**	5.90
Cert. PBB $\epsilon = 0.035$	16.67	

The indicated error rate on CIFAR-10 is much lower than the one in [PORSTS21]. We attribute this to ResNet18 being a stronger feature extractor than the 15-layer CNN used in [PORSTS21]. By comparison with the stochastic prediction error rate, risk certificates are also very tight. This is roughly on par with the results of [PORSTS21] which improves on earlier work by [DR18].

Our contribution is not to find tighter risk certificates, but to provide a novel method that enables a more computationally efficient way to compute them. Because pushing forward an intrinsic normal distribution of initial LDAF assignment states is only about c times more expensive than an LDAF forward pass, sampling the integrand in (41b) is very cheap by comparison to earlier works, which require a separate forward pass per sample. This allows to, e.g., compute the reference solution used in Figure 2 in 14 minutes on a single GPU for 100M Monte-Carlo samples and batch size 100.

We use cross-entropy as a differentiable surrogate loss for training PAC-Bayes posteriors. This appears problematic because the bound (7) only certifies risk w.r.t. bounded loss functions. [PORSTS21] address this by modifying cross-entropy to obtain a closely related bounded loss function which is amenable to risk certification. We do not perform this modification and therefore do not obtain valid risk certificates for surrogate loss. However, for classification (01 loss) the bound (7) holds for *all* posterior distributions, regardless of how they have been computed. Therefore, using unbounded

surrogate loss for training does not touch the validity of risk certificates for the bounded 01-loss reported in Table 1. Accordingly, no certificate for surrogate loss is reported.

A key component of the proposed approach are linearized deep assignment flows (LDAFs). We view them as uniquely suitable due to the combination of two factors. (1) The pushforward of normal distributions under LDAF dynamics has a closed form and efficient numerics exist to approximate its moments. (2) Unlike trivial maps which have the first property, the LDAF still has nontrivial representational power.

To illustrate this, attempt to replace the LDAF within the given framework by a learned dense linear operator. Aside from potentially introducing a large number of additional parameters, this effectively merely adds noise to the classification logits and only the stochastic classifier mean depends on data. Thus, essentially no improvement of the posterior over the prior is to be expected. By contrast, the low-rank numerics used to compute the pushforward under the LDAF reveal additional information about learnt parameters that we will exploit in future work.

Acknowledgements

This work is funded by the Deutsche Forschungsgemeinschaft (DFG), grant SCHN 457/17-1, within the priority programme SPP 2298: "Theoretical Foundations of Deep Learning"

This work is funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy EXC-2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster).

References

[AJVLS17] Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer, *Information geometry*, vol. 64, Springer International Publishing, 2017.

[ÅPSS17] Freddie Åström, Stefania Petra, Bernhard Schmitzer, and Christoph Schnörr, *Image Labeling by Assignment*, Journal of Mathematical Imaging and Vision **58** (2017), no. 2, 211–238.

[BCKW15] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, *Weight Uncertainty in Neural Network*, International Conference on Machine Learning, vol. 37, PMLR, July 2015, pp. 1613–1622.

[BO16] Kinjal Basu and Art B. Owen, *Transformations and Hardy–Krause Variation*, SIAM Journal on Numerical Analysis **54** (2016), no. 3, 1946–1966 (English).

[BSS21] Bastian Boll, Jonathan Schwarz, and Christoph Schnörr, *On the Correspondence Between Replicator Dynamics and Assignment Flows*, Proceedings SSVM, LNCS, vol. 12679, Springer International Publishing, 2021, pp. 373–384.

[Cat07] Oliver Catoni, *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, IMS Lecture Notes Monograph Series, vol. 56, Institute of Mathematical Statistics, 2007.

[CRBD18] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud, *Neural Ordinary Differential Equations*, NeurIPS, 2018.

[DKS13] Josef Dick, Frances Y. Kuo, and Ian H. Sloan, *High-Dimensional Integration: The Quasi-Monte Carlo Way*, Acta Numerica **22** (2013), 133–288.

[DR18] Gintare Karolina Dziugaite and Daniel M. Roy, *Data-Dependent PAC-Bayes Priors via Differential Privacy*, NeurIPS, 2018.

[GAZS22] Daniel Gonzalez-Alvarado, Alexander Zeilmann, and Christoph Schnörr, *Quantifying Uncertainty of Image Labelings Using Assignment Flows*, DAGM GCPR: Pattern Recognition, Lecture Notes in Computer Science, vol. 13024, Springer International Publishing, January 2022, pp. 453–466.

[Gue19] Benjamin Guedj, *A Primer on PAC-Bayesian Learning*.

[HO10] Marlis Hochbruck and Alexander Ostermann, *Exponential Integrators*, Acta Numerica **19** (2010), 209–286.

[HS03] Josef Hofbauer and Karl Siegmund, *Evolutionary Game Dynamics*, Bulletin of the American Mathematical Society **40** (2003), no. 4, 479–519.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[KH$^+$09] Alex Krizhevsky, Geoffrey Hinton, et al., *Learning multiple layers of features from tiny images*.

[LC02] John Langford and Rich Caruana, *(Not) Bounding the True Error*, Advances in Neural Information Processing Systems, vol. 14, MIT Press, 2002, pp. 809–816.

[LH16] Ilya Loshchilov and Frank Hutter, *Sgdr: Stochastic gradient descent with warm restarts*, arXiv preprint arXiv:1608.03983 (2016).

[LS01] John Langford and Matthias Seeger, *Bounds for Averaging Classifiers*, Technical Report CMU-CS-01-102 (2001).

[MC95] William J. Morokoff and Russel E. Caflisch, *Quasi-Monte Carlo Integration*, Journal of Computational Physics **122** (1995), 218–230.

[MM15] Dario Madeo and Chiara Mocenni, *Game Interactions and Dynamics on Networked Populations*, IEEE Transactions on Automatic Control **60** (2015), no. 7, 1801–1810.

[NW12] Jitse Niesen and Will M. Wright, *Algorithm 919: A Krylov Subspace Algorithm for Evaluating the φ-Functions Appearing in Exponential Integrators*, ACM Transactions on Mathematical Software **38** (2012), no. 3, 1–19.

[PORSTS21] Maria Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári, *Tighter Risk Certificates for Neural Networks*, Journal of Machine Learning Research **22** (2021), no. 227, 1–40.

[Pri58] Robert Price, *A useful theorem for nonlinear devices having Gaussian inputs*, IRE Transactions on Information Theory **4** (1958), no. 2, 69–72.

[RH05] Havard Rue and Leonhard Held, *Gaussian markov random fields: Theory and applications*, Monographs on statistics and applied probability, no. 104, Chapman & Hall/CRC, February 2005.

[Saa92] Yousef Saad, *Analysis of Some Krylov Subspace Approximations to the Matrix Exponential Operator*, SIAM Journal on Numerical Analysis **29** (1992), no. 1, 209–228.

[Sch20] Christoph Schnörr, *Assignment Flows*, Handbook of Variational Methods for Nonlinear Geometric Data (P. Grohs, M. Holler, and A. Weinmann, eds.), Springer International Publishing, 2020, pp. 235–260.

[SS21] F. Savarino and C. Schnörr, *Continuous-Domain Assignment Flows*, European Journal of Applied Mathematics **32** (2021), no. 3, 570–597.

[TIWS17] Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin, *A Strongly Quasiconvex PAC-Bayesian Bound*, Int. Conf. Algorithmic Learning Theory (ALT), vol. 76, PMLR, October 2017, pp. 466–492.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*, 2017.

[ZK16] Sergey Zagoruyko and Nikos Komodakis, *Wide residual networks*, Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, September 2016, pp. 87.1–87.12.

[ZPS21] Alexander Zeilmann, Stefania Petra, and Christoph Schnörr, *Learning Linear Assignment Flows for Image Labeling via Exponential Integration*, Proceedings SSVM, vol. 12679, Springer International Publishing, 2021, pp. 385–397.

[ZSPS20] Alexander Zeilmann, Fabrizio Savarino, Stefania Petra, and Christoph Schnörr, *Geometric Numerical Integration of the Assignment Flow*, Inverse Problems **36** (2020), no. 3, 034004.

[ZZS21] Artjom Zern, Alexander. Zeilmann, and Christoph Schnörr, *Assignment Flows for Data Labeling on Graphs: Convergence and Stability*, Information Geometry (2021).

Appendix A. ResNet Training

The ResNet18 architecture was proposed by [HZRS16] for classification on ImageNet which contains much larger images than the ones considered here. In order to account for small image dimensions (CIFAR: 32×32 pixels, FashionMNIST: 28×28 pixels), we swap the first convolution in ResNet18 (stride 2, kernel size 7) for a smaller one (stride 1, kernel size 3). We train with randomly initialized weights and do not change the training regime between CIFAR-10 and FashionMNIST. Our feature extractors and reference classifier implementation are based on the freely available PyTorch implementation `https://github.com/kuangliu/pytorch-cifar`.

Image and Pattern Analysis Group, Heidelberg University, Germany
Email address: `bastian.boll@iwr.uni-heidelberg.de`
URL: `https://ipa.math.uni-heidelberg.de`