Received: 7 October 2022 | Accepted: 21 December 2022

DOI: 10.1002/pamm.202200169

Self-Certifying Classification by Linearized Deep Assignment

Bastian Boll[1,*], **Alexander Zeilmann**[1], **Stefania Petra**[2], and **Christoph Schnörr**[1]

[1] Image and Pattern Analysis Group, Heidelberg University, Heidelberg

[2] Mathematical Imaging Group, Heidelberg University, Heidelberg

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.

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 [1, 2] has attracted much attention due to the recent demonstration of tight risk bounds for deep stochastic neural networks in [3]. 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. The recent work [4] evaluates various *relaxed PAC-Bayes-kl inequalities*[1] [5], including a new one. They show 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 allows to use the whole data set for both learning a predictor and certifying its risk. Similar to [4], our approach is to find a PAC-Bayes posterior distribution by optimizing the PAC-Bayes-λ inequality introduced by [6]. 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. [7] 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. [4] choose the pathwise gradient estimator [8] and call the resulting framework PAC-Bayes with Backprop, reminiscent of the Bayes-by-Backprop paradigm [9]. 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* [10]. After suitable parametrization and linearization, the uncertainty quantification approach proposed in [11] allows to push forward intrinsic normal distributions of initial assignment states in closed form, which can be leveraged to build deep stochastic classifiers with tractable empirical risk.

1.2 Contribution

We adopt the PAC-Bayes-λ inequality [6] to work out a two-stage method as in [3, 4] 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 [12] and compute the corresponding pushforward distribution in closed form, building on [11]. This enables 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 [13] can replace Monte-Carlo estimation when evaluating risk certificates and computing their gradients. As a consequence, the *linearized deep assignment flow approach* to classification becomes a *self-certifying learning method*. We verify its performance and the tightness of risk certificates by a comparison to the empirical test error.

* Corresponding author: e-mail bastian.boll@iwr.uni-heidelberg.de

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

2 Background

2.1 (S-)Assignment Flows

The *assignment flow approach* [10, 14] denotes a class of dynamical systems for analyzing metric data on a graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, $|I| = n$, 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. For the *classification* task considered here, the statistical manifold is the relative interior $\mathcal{S}_c$ of probability simplex with c vertices equipped with the Fisher-Rao metric of information geometry [15], and the interaction corresponds to *geometric* state averaging derived from the affine e-connection. The resulting dynamical system reads

$$\dot{S}(t) = R_{S(t)}[\Omega S(t)], \quad S(0) = S_0 \tag{2.1a}$$

$$\text{where} \quad \dot{S}_i(t) = R_{S_i(t)}(Z_i(t)) = \operatorname{Diag}(S_i)Z_i - \langle S_i(t), Z_i(t)\rangle S_i(t), \qquad Z_i(t) = (\Omega S(t))_i \,. \tag{2.1b}$$

Here $S(t) \in \mathcal{W} \subset \mathbb{R}^{n\times c}_{++}$ comprises the state at each vertex $i \in \mathcal{I}$ as row vector $S_i(t) \in \mathcal{S}_c$, $2 \leq c \in \mathbb{N}$ denotes the number of classes, and $\mathcal{W}$ is the n-fold product of $\mathcal{S}_c$. Ω is a weighted adjacency matrix of $\mathcal{G}$ and R_S defined in (2.1b) is called the replicator operator. Thus, (2.1) may be seen as a particular system of neural ODEs [16] 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 (2.1) and restrict ourselves to a *linearization* of the resulting large-scale dynamical system.

2.2 PAC-Bayes Risk Certification

Consider stochastic classifiers, i.e., distributions μ over a hypothesis space $\mathcal{H}$, elements of which are functions ϕ_θ mapping a data space $\mathcal{D}$ to the tangent space $\mathcal{T}_0$ of $\mathcal{W}$. 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

$$\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}(\theta)] := \mathbb{E}_{\theta\sim\mu}\big[\mathbb{E}_{(x,y)\sim\mathfrak{D}}\ell(\phi_\theta(x), y)\big] \tag{2.2}$$

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* $\mathfrak{L}_m(\theta)$ which replaces the inner expectation in (2.2) by a mean over m i.i.d. samples (x_k, y_k) drawn from $\mathfrak{D}$. PAC-Bayesian theory [1, 2] 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 has access to *fewer* data. A goal is to construct tight, high-confidence bounds on (2.2) which only depend on tractable quantities such as empirical risk. In our analysis, we use the following state-of-the-art bound.

Theorem 2.1 (**PAC-Bayes-λ Inequality** [6]) *For any $\epsilon > 0$ and any $\lambda \in (0, 2)$, it holds with probability at least $1 - \epsilon$ over the i.i.d. sample of size m for all posterior distributions μ over parameters θ simultaneously*

$$\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}(\theta)] \leq \frac{\mathbb{E}_{\theta\sim\mu}[\mathfrak{L}_m(\theta)]}{1 - \frac{\lambda}{2}} + \frac{\mathrm{KL}(\mu\colon \pi) + \log\frac{2\sqrt{m}}{\epsilon}}{m\lambda(1 - \frac{\lambda}{2})}, \tag{2.3}$$

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 Classification Using Deep S-Flows

Motivated by the use of coupled replicator dynamics in game theory [17], we generalize S-flows (2.1) by enabling additional interaction on the label space. Specifically, we consider the vectorized version of the assignment flow equation (2.1)

$$\dot{s}(t) = R^{\mathfrak{v}}_{s(t)}(\Omega \otimes \mathbb{I}_c)s(t), \qquad s(0) = s_0 = \operatorname{vec}(S_0) \tag{3.1}$$

and break up the Kronecker product structure of $\Omega \otimes \mathbb{I}_c$. Re-using the symbol Ω to denote a matrix $\Omega \in \mathbb{R}^{N\times N}$, $N = cn$ we define the *deep assignment flow* (DAF) in vectorized form as

$$\dot{s}(t) = R^{\mathfrak{v}}_{s(t)}\Omega s(t), \qquad s(0) = s_0. \tag{3.2}$$

This class of dynamics is more general than (2.1), while remaining amenable to lifting and linearization with minimal modifications to other assignment flows [18, 19]. Concerning the PAC-Bayes risk certification, we observe that (3.2) typically

leads to better generalization and more gain between posterior and prior as compared to (2.1). 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 $\mathcal{G}$ to be relatively small ($n = 50$ nodes) and densely connected with learned symmetric matrix Ω. We also designate a single node to carry class probabilities. Through the dynamics (3.2), the state of this node will evolve towards an integer assignment, i.e., a class decision. By convention, we choose the classification node be the node with index 1 and set $s_{[c]} = S_1 \in \mathcal{S}_c$, i.e. $s_{[c]}(t)$ is the subvector of the solution $s(t) = \mathrm{vec}^{-1}(S(t))$ to the DAF (3.2) corresponding to the assignment vector S_1 indexed by the first vertex $i = 1 \in \mathcal{I}$ of the underlying graph.

3.2 Linearized Deep Assignment Flows

A key technical ingredient of our contribution concerns the following approximation of the DAF (3.2) that evolves on the manifold $\mathcal{W}$, obtained by a *linear* parametrization on the tangent space $\mathcal{T}_0$.

Proposition 3.1 (**Linearized deep assignment flow (LDAF)**) *The system of equations*

$$s(t) = \exp_{s_0}^{\mathfrak{v}}(v(t)), \qquad \dot{v}(t) = \Pi_0^{\mathfrak{v}}\Omega(s_0 + R_{s_0}^{\mathfrak{v}}v(t)), \qquad v(0) = v_0 = 0 \tag{3.3}$$

closely approximates the deep assignment flow (3.2)*. The solution is given in closed form by*

$$v(t) = t\varphi(tA)v_D, \qquad A = \Pi_0^{\mathfrak{v}}\Omega R_{s_0}^{\mathfrak{v}}, \quad v_D = \Pi_0^{\mathfrak{v}}\Omega s_0, \tag{3.4}$$

where φ is the analytical matrix function $\varphi(z) = \frac{e^z-1}{z}$ with matrix argument, $\exp_{s_0}^{\mathfrak{v}} v = $ softmax$(v + \log s_0)$ and $\Pi_0^{\mathfrak{v}}$ denotes orthogonal projection to the tangent space.

The solution $s(t)$ to the linearized deep assignment flow (LDAF) (3.3) can be efficiently solved using Krylov methods for evaluating (3.4). Moreover, gradient approximations computed in [20] apply without modification. We point out that even though $\varphi(tA)$ acts linearly on v_D in (3.4), LDAF dynamics are nonlinear models. 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 [7]. 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 (3.3), (3.4) up to time $T > 0$. 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 $\mathcal{D}$, 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 $\mathcal{H} = \{\phi \colon \mathbb{R}^d \to \mathcal{S}_c \mid \phi = \psi_{\omega,v_0} \circ \exp_{\mathbb{1}_\mathcal{W}} \circ F_\vartheta\}$ 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

$$\mu = \delta_\vartheta \times \delta_\omega \times \mathcal{N}(0, \Sigma_0) \tag{4.1}$$

where $\mathcal{N}(0, \Sigma_0)$ denotes an intrinsic normal distribution on $\mathcal{T}_0$ (cf. chapter 3 in [21]) centered at 0. 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 (4.1). Further, ω and ϑ are fixed and only the distribution of v_0 differs between π and μ (PAC-Bayes posterior). This ensures that the posterior is absolutely continuous with respect to the prior ($\mu \ll \pi$) which makes their relative entropy well-defined.

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 [3]. 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.

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 [11] to the deep flows considered here.

Proposition 4.1 (**LDAF Pushforward**) *Consider the LDAF dynamics* (3.3) *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*

$$\mathfrak{m}(t) = t\varphi(tA)b, \qquad \Sigma(t) = \operatorname{expm}(tA)\Sigma_0 \operatorname{expm}(tA)^\top \tag{4.2}$$

Proof. For $v_0 \neq 0$, the closed form solution (3.4) is modified to $v(t) = \operatorname{expm}(tA)v_0 + t\varphi(tA)b$. 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. Further, analogous to the computation in [11] one finds the moments (4.2). □

The full covariance matrix $\Sigma(t) \in \mathbb{R}^{N\times N}$ is quite large ($N = nc$) 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. We may now leverage the available closed form (4.2) to transform the empirical risk of stochastic LDAF classifiers, which is the main technical contribution of this paper.

Theorem 4.2 (**LDAF Expected Empirical Risk**) *Fix (linear) coordinates of* $T_0\mathcal{S}_c$ *by choosing the columns of*

$$P := \begin{pmatrix} \mathbb{I}_{c-1} \\ -\mathbb{1}_{c-1}^\top \end{pmatrix} \in \mathbb{R}^{c\times(c-1)} \tag{4.3}$$

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*

$$\frac{1}{m}\sum_{k\in[m]} \int_{\mathbb{R}^{c-1}} \ell(Pz + F_\vartheta(x_k), y_k)\rho_{\widehat{\mathfrak{m}}(x_k),\widehat{\Sigma}(x_k)}(z)\mathrm{d}z. \tag{4.4}$$

Here, ρ *denotes the density of a multivariate normal distribution with the indicated moments* $\widehat{\mathfrak{m}}(x_k) = \mathfrak{m}(T)_{\mathcal{I}\setminus\{c\}}$ *and* $\widehat{\Sigma}(x_k) = \Sigma(T)_{\mathcal{I}\setminus\{c\},\mathcal{I}\setminus\{c\}}$ *which are subvectors resp. submatrices of* (4.2) *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* (4.3).

Proof. We use Proposition 4.1 to transform the expected empirical risk integral. The basis (4.3) is chosen such that the sought moments at the classification node can be selected as entries of pushforward moments. □

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 4.2 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. 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 [13][Theorem 3.9], 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. In practice, QMC methods are observed to outperform MC particularly for moderate dimension and smooth integrands [22]. 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 1.

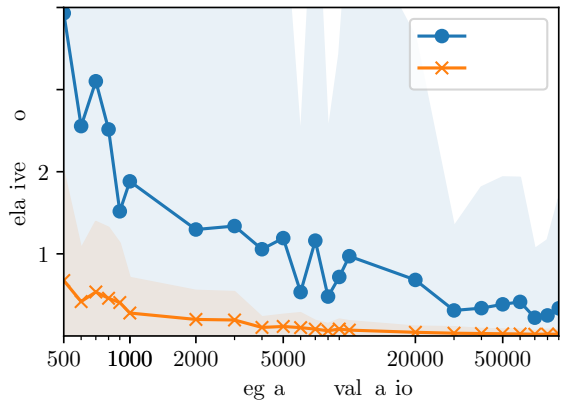

Fig. 1 Accuracy of QMC integration ■ and MC integration ■ for computing the expected empirical risk with varying number of sample points. Error bands indicate standard deviation within a batch of 100 CIFAR-10 data points. Because the pushforward distribution of Theorem 4.2 is computationally tractable, 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.

	Deterministic	Prior	Posterior	Certificate	Tightness
LDAF	20.54 ± 0.58	**20.89**	**20.46**	**21.55**	**1.09**
PBB [4]	**19.46**	21.69	20.81	23.77	2.96

Table 1 Out-of-sample CIFAR-10 classification error (%) and risk certificates compared to [4] (PBB). Certificates ($\epsilon = 0.035$) computed using 150k MC samples for PBB (multiple GPU hours) and 100k QMC samples for LDAF ($\sim$ 90 GPU seconds).

	Deterministic	Prior	Posterior	Cert. ($\epsilon = 0.01$)	Cert. ($\epsilon = 0.05$)
CIFAR-10	5.28 ± 0.06	5.49	5.31	6.36	6.19
FashionMNIST	5.13 ± 0.16	5.13	5.12	6.07	5.90

Table 2 Out-of-sample error (%) and risk certificates of LDAF classifiers with ResNet18 features on CIFAR-10 and FashionMNIST.

5 Benchmarks and Discussion

As empirical support for the applicability of the proposed self-certifying approach to classification, we perform image classification on CIFAR-10 [23] and FashionMNIST [24].

5.1 Training Stochastic LDAF Classifiers

Stochastic classifiers π and μ are implemented as distributions with shape (4.1) over the hypothesis space $\mathcal{H}$. The graph G is chosen relatively small ($n = 50$ nodes) and densely connected with symmetric adjacency matrix Ω. 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 diagonal matrix plus rank-one update. To train stochastic classifiers, we proceed in two steps. First, 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. Initializing μ at π, we subsequently train μ by minimizing the r.h.s. of the bound (2.3), alternating between optimization of μ and λ after each epoch on the validation set.

For direct comparison with [4] on CIFAR-10 (Table 1), we use the same 9-layer CNN feature extractor and a simple SGD training regime (70 epochs, learning rate 0.01, momentum 0.95, dropout rate 0.2) without data augmentation. Our deterministic classifier performs slightly worse, likely due to a lack of hyperparameter tuning, while stochastic classifiers built on the same features within the proposed framework slightly outperform the related work. By comparing to posterior test set error, we find that our risk certificate is slightly tighter in this particular benchmark. However, *our main contribution is* not to improve tightness, but *to provide a novel method that enables a more computationally efficient way to compute risk certificates*. Because pushing forward an intrinsic normal distribution of initial LDAF assignment states is only marginally more expensive than an single forward pass, sampling the empirical risk integrand is very cheap ($\mathcal{O}(m) = 15$k forward passes) and can be realized within few minutes on a single GPU. By comparison, drawing 150k MC samples to compute the certificate in the framework of [4] requires many GPU hours for a 9-layer CNN model ($\mathcal{O}(m \cdot q) = 2.25$B forward passes). In addition, fewer samples are required when opting for QMC integration.

For *deeper models*, more computational effort is required per forward pass, leading to *even larger runtime gains by choosing the proposed framework*. To illustrate scalability, we chose ResNet18 [25] features with an adapted training and light data augmentation regime as in [26]. The resulting feature extractors are much stronger, leading to higher classification scores in Table 2 while still *achieving tight, high confidence risk certificates very efficiently*.

5.2 Discussion and Conclusion

We use cross-entropy as a differentiable surrogate loss for training PAC-Bayes posteriors. This appears problematic because the bound (2.3) only certifies risk w.r.t. bounded loss functions. [4] 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 (0/1 loss) the bound (2.3) 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 0/1 loss reported in Tables 1 and 2. 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. In addition, the low-rank numerics used to compute the pushforward under the LDAF reveal information about learned 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). Open access funding enabled and organized by Projekt DEAL.

References

[1] O. Catoni, PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning, IMS Lecture Notes Monograph Series, Vol. 56 (Institute of Mathematical Statistics, 2007).
[2] B. Guedj, A primer on PAC-Bayesian learning, in: Proceedings of the second congress of the French Mathematical Society, French Mathematical Society, Vol. 33 (French Mathematical Society, 2019).
[3] G. K. Dziugaite and D. M. Roy, Data-dependent PAC-Bayes priors via Differential Privacy, in: Advances in Neural Information Processing Systems, NIPS, Vol. 31 (Curran Associates, Inc., 2018).
[4] M. Pérez-Ortiz, O. Rivasplata, J. Shawe-Taylor, and C. Szepesvári, Tighter Risk Certificates for Neural Networks, Journal of Machine Learning Research **22**(227), 1–40 (2021).
[5] J. Langford and M. Seeger, Bounds for Averaging Classifiers, Technical Report CMU-CS-01-102 (2001).
[6] N. Thiemann, C. Igel, O. Wintenberger, and Y. Seldin, A Strongly Quasiconvex PAC-Bayesian Bound, in: Proceedings of the 28th International Conference on Algorithmic Learning Theory, Proceedings of Machine Learning Research, Vol. 76 (PMLR, 2017), pp. 466–492.
[7] J. Langford and R. Caruana, (Not) Bounding the True Error, in: Advances in Neural Information Processing Systems, NIPS, Vol. 14 (MIT Press, 2001).
[8] R. Price A useful theorem for nonlinear devices having Gaussian inputs, IRE Transactions on Information Theory **4**(2), 69–72 (1958).
[9] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, Weight uncertainty in neural network, in: Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 37 (PMLR, Lille, France, 2015), pp. 1613–1622.
[10] F. Åström, S. Petra, B. Schmitzer, and C. Schnörr, Image Labeling by Assignment, Journal of Mathematical Imaging and Vision **58**(2), 211–238 (2017).
[11] D. Gonzalez-Alvarado, A. Zeilmann, and C. Schnörr, Quantifying Uncertainty of Image Labelings Using Assignment Flows, in: DAGM GCPR: Pattern Recognition, LNCS, Vol. 13024 (Springer, 2022), pp. 453–466.
[12] F. Savarino and C. Schnörr, Continuous-Domain Assignment Flows, European Journal of Applied Mathematics **32**(3), 570–597 (2021).
[13] J. Dick, F. Y. Kuo, and I. H. Sloan, High-Dimensional Integration: The Quasi-Monte Carlo Way, Acta Numerica **22**, 133–288 (2013).
[14] C. Schnörr, Assignment Flows, in: Handbook of Variational Methods for Nonlinear Geometric Data, edited by P. Grohs, M. Holler, and A. Weinmann (Springer, 2020), pp. 235–260.
[15] N. Ay, J. Jost, H. V. Lê, and L. Schwachhöfer, Information Geometry, A Series of Modern Surveys in Mathematics, Vol. 64 (Springer, Cham, 2017).
[16] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Neural ordinary differential equations, in: Advances in Neural Information Processing Systems, NIPS, Vol. 31 (Curran Associates, Inc., 2018).
[17] D. Madeo and C. Mocenni, Game Interactions and Dynamics on Networked Populations, IEEE Transactions on Automatic Control **60**(7), 1801–1810 (2015).
[18] A. Zeilmann, F. Savarino, S. Petra, and C. Schnörr, Geometric Numerical Integration of the Assignment Flow, Inverse Problems **36**(3), 034004 (2020).
[19] B. Boll, J. Schwarz, and C. Schnörr, On the Correspondence Between Replicator Dynamics and Assignment Flows, in: Proceedings SSVM, LNCS, Vol. 12679 (Springer, 2021), pp. 373–384.
[20] A. Zeilmann, S. Petra, and C. Schnörr, Learning linear assignment flows for image labeling via exponential integration, in: Proceedings SSVM, LNCS, Vol. 12679 (Springer, 2021), p. 385–397.
[21] H. Rue and L. Held, Gaussian Markov Random Fields: Theory and Applications, No. 104 in Monographs on statistics and applied probability (Chapman & Hall/CRC, 2005).
[22] W. J. Morokoff and R. E. Caflisch, Quasi-Monte Carlo Integration, Journal of Computational Physics **122**, 218–230 (1995).
[23] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto, Toronto, Ontario, 2009.
[24] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017.
[25] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (Institute of Electrical and Electronics Engineers (IEEE), 2016), pp. 770–778.
[26] S. Zagoruyko and N. Komodakis, Wide Residual Networks, in: Proceedings of the British Machine Vision Conference (BMVC), (BMVA Press, 2016), pp. 87.1–87.12.

