On PAC-Bayes Bounds for Deep Neural Networks using the Loss Curvature

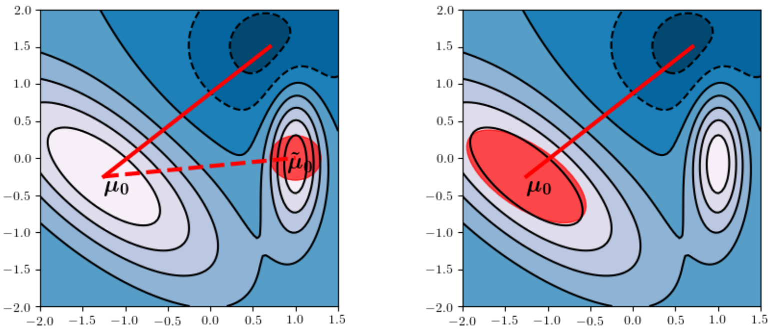
Anonymous authors
Paper under double-blind review

Abstract

We investigate whether it's possible to tighten PAC-Bayes bounds for deep neural networks by utilizing the Hessian of the training loss at the minimum. For the case of Gaussian priors and posteriors we introduce a Hessian-based method to obtain tighter PAC-Bayes bounds that relies on closed form solutions of layerwise subproblems. We thus avoid commonly used variational inference techniques which can be difficult to implement and time consuming for modern deep architectures. We conduct a theoretical analysis that links the random initialization, minimum, and curvature at the minimum of a deep neural network to limits on what is provable about generalization through PAC-Bayes. Through careful experiments we validate our theoretical predictions and analyze the influence of the prior mean, prior covariance, posterior mean and posterior covariance on obtaining tighter bounds.

1 Introduction

Deep neural networks are by now the established method for tackling a number of machine learning tasks. Despite this their performance on out of sample data is very difficult to be proven formally and is usually validated empirically by using a set of validation samples. Classic measures of capacity such as the VC dimension which are uniform across all functions representable by the classification architecture are doomed to fail; DNNs are typically overparameterized and correspondingly the set of representable functions is large enough to make the the bounds vacuous. For example in the



(a)

(b)

Figure 1: **The importance of retaining the original minimum**: Conventional wisdom links tight generalization error bounds to correctly estimating flat and curved directions in the loss landscape around the original minimum $\boldsymbol{\mu}_0$. Existing optimization based non-vacuous bounds compute implicitly or explicitly a different minimum $\tilde{\boldsymbol{\mu}}_0$ and then implicitly evaluate the curvature and posterior distribution around that minimum. By contrast we aim to estimate the curvature and a related optimal posterior distribution around the original solution $\boldsymbol{\mu}_0$.

highly cited work Zhang et al. (2016) the authors show that the set of representable functions of typical DNNs contains elements that can memorize the labels over the training set.

A lesson drawn from this experiment is that defining the hypothesis class a priori results in bounds that are too loose. Clearly from empirical observation optimization algorithms reach solutions that are not trivially memorizing the labels. As such recently researchers turned to measures of complexity that are data dependent and are defined a posteriory; that is taking into account the specific solution achieved after optimization. One way of achieving this is by defining bounds that incorporate norms of the learned layer weights. Examples include Bartlett et al. (2017) Neyshabur et al. (2017) Golowich et al. (2017) and have been reached through a number of proof techniques. However these analytical bounds when evaluated explicitly are still vacuous by several orders of magnitude. They have been motivated simply by empirical correlations with generalization error; an argument which has been criticized in a number of works Kawaguchi et al. (2017) Nagarajan & Kolter (2019b) Pitas et al. (2019).

On a more fundamental level simple correlation with generalization error is unsatisfying for more critical applications such as healthcare, autonomous driving, and finance where DNNs are increasingly being deployed, potentially making life-altering decisions. Consequently some works have achieved success in proving generalization in specific settings by optimizing PAC-Bayesian bounds McAllester (1999). PAC-Bayes theorems typically assume a randomized classifier defined by a posterior distribution Q, they then bound the generalization error of this randomized classifier by using as a measure of complexity the KL-Divergence between the posterior distribution Q and a proper prior distribution P. The prior P is meant to model a ”very simple function” and is usually chosen to be a scaled standard Gaussian distribution. In Dziugaite & Roy (2017) the authors optimize the posterior distribution while enforcing non-trivial training accuracy so as to obtain a non-vacuous bound on significantly simplified MNIST datasets. In Zhou et al. (2018) the authors compress an original neural network therefore minimizing it’s effective capacity while constraining it to have high accuracy over the training set. The obtained network can be proven to have non-vacuous generalization even for large scale Imagenet experiments. Both results remain significantly loose.

It is worthwhile to note the subtle but important ways in which the above two works diverge from PAC-Bayesian intuition. PAC Bayes defines an a posteriori hypothesis class roughly as a ball around the obtained classifier solution, this ball is defined implicitly by assuming a posterior that is usually a Gaussian with a given variance. The larger the variance of the posterior, the larger the ball that can be placed on the obtained solution and the simpler the hypothesis class, or in the case of derandomized PAC-Bayes the simpler the individual hypothesis. By optimizing the mean of the posterior in Dziugaite & Roy (2017) and by applying compression in Zhou et al. (2018) the authors arrive to posteriors whose weights are not similar even in expectation to the original classifier. Furthermore intuition regarding the role of the magnitude of the variance, is largely destroyed. At the same time neither the non-convex optimization problem solved in Dziugaite & Roy (2017) nor the compression schemes employed in Zhou et al. (2018) are guaranteed to converge to a global minimum. The Variational Inference objective employed in Dziugaite & Roy (2017) is especially difficult to optimize and has hindered the widespread adoption of Bayesian neural networks Wu et al. (2018). It is therefore an open problem to test the limits of PAC-Bayes for proving generalization in the original classifiers obtained by optimization, while ideally avoiding Variational Inference.

In Dziugaite & Roy (2018) the authors take a step in this direction by optimizing the prior of the PAC-Bayes bound. PAC-Bayesian theory allows the prior to be distribution dependent but not dependent on the training set, the authors enforce this constraint through the differential privacy approach Dwork (2011). Both objectives in Dziugaite & Roy (2017), Dziugaite & Roy (2018) are however difficult to optimize for anything but small scale experiments.

Our work has close connections with Achille & Soatto (2018)Achille et al. (2019). These works aim to link on a fundamental level the Kolmogorov, Shannon and Fisher Information in deep neural networks to the sufficiency,minimality, and invariance of their representations. Our work by contrast focuses on tightening PAC-Bayesian bounds and determining how much progress can be made towards non-vacuous bounds simply by leveraging local properties of a given minimum.

We adopt the PAC-Bayesian approach and seek to find optimal solutions that circumvent the above mentioned Variational Inference problems. At the same time we want to clarify the contribution of the prior mean and covariance choices in obtaining non vacuous bounds. We focus on the popular choice of multivariate Gaussians with diagonal covariance and fixed means to model priors and posteriors.

2 CONTRIBUTIONS

- We use a second order Taylor expansion of the randomized empirical loss in an IB-Lagrangian objective. Using this approximation we derive lower bounds in the IB-Lagrangian objective that correspond to an invalid but optimal PAC-Bayes bound (the prior is training set dependent).
- We propose a theoretically motivated layerwise method to obtain optimal PAC-Bayes bounds with respect to the posterior variance, that are also valid (the prior is non-informative).
- We conduct experiments where the valid bounds using our method approach the invalid optimal ones. In other cases we can rule out being able to prove generalization using our modeling assumptions. We also conduct a detailed analysis of the contribution of the prior mean and covariance in obtaining non-vacuous bounds.

3 PAC-BAYESIAN FRAMEWORK

We consider the hypothesis class $\mathcal{H}_L$ realized by the feedforward neural network architecture of depth L with coordinate-wise activation functions σ defined as the set of functions $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ $(\mathcal{X} \subseteq \mathbb{R}^p, \mathcal{Y} \subseteq \mathbb{R}^K)$ with $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\sigma(...\sigma(\boldsymbol{x}^T\mathbf{W}_0)\mathbf{W}_1)\mathbf{W}_2)..)\mathbf{W}_L)$ where $\boldsymbol{\theta} \in \Theta_L \subseteq \mathbb{R}^d$ is a vectorization of the weights and $\Theta_L = \mathbb{R}^{p\times k_1} \times \mathbb{R}^{k_1\times k_2} \times ... \times \mathbb{R}^{k_L \times K}$. Given the loss function $\ell(\cdot,\cdot)$ we can define the population loss: $L(\boldsymbol{\theta}) := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{P}}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})$ and given a training set of N instances $S = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^N$ the empirical loss $\hat{L}(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^N \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)$.

The PAC-Bayesian framework McAllester (1999) provides generalization error guarantees for randomized classifiers drawn from a posterior distribution Q. We will use the following form of the PAC-Bayes bound.

Theorem 3.1. *(PAC-Bayesian theorem McAllester (1999)) For any data distribution over* $\mathcal{X} \in \{-1,+1\}$*, we have that the following bound holds with probability at least* $1-\delta$ *over random i.i.d. samples* $S = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^N$ *of size N drawn form the data distribution:*

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[L(\boldsymbol{\theta})] \leq \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{L}(\boldsymbol{\theta})] + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{2(N-1)}{\delta}}{2N}}. \tag{1}$$

Here Q is an arbitrary "posterior" distribution over parameter vectors, which may depend on the sample S and on the prior P.

The framework models the complexity of the randomized classifier as the KL-Divergence between the posterior Q and a prior P. The prior P must be valid in that it cannot depend in any way on the training data. On the contrary the posterior Q can be chosen to be any arbitrary distribution. This flexibility allows one to model deterministic neural networks as the mean of an arbitrary posterior distribution, thus deriving results for a stochastic but closely related classifier.

3.1 PREVIOUS WORK

As the analytical solution for the KL term in 1 obviously underestimates the noise robustness of the deep neural network around the minimum one might be tempted to obtain a tighter PAC-Bayes bound by directly optimizing

$$\mathcal{L}(Q(w|\mathcal{D})) = \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{L}(\boldsymbol{\theta})] + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{2(N-1)}{\delta}}{2N}} \tag{2}$$

so as to obtain a posterior that is both close to the PAC-Bayesian prior and has a non-vacuous accuracy. Optimizing the above objective cannot be done directly as computing $\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{L}(\boldsymbol{\theta})]$ or it's gradients is intractable for general distributions Q. A typical workaround is to parameterize $\boldsymbol{\theta}$ as having a Gaussian distribution $\boldsymbol{\theta} = \boldsymbol{\mu} + \boldsymbol{\xi} \odot \boldsymbol{\sigma}$ where $\boldsymbol{\xi} \sim \mathcal{N}(0, I)$ and compute gradients of the resulting unbiased estimate $\mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{N}(0,I)}[\hat{L}(\boldsymbol{\mu} + \boldsymbol{\xi} \odot \boldsymbol{\sigma})]$. In Dziugaite & Roy (2017) the authors use this technique and then present a non-vacuous bound for fully connected deep neural networks.

Remarkably the above formulation bears striking resemblance to the objective

$$C_\beta(\mathcal{D}; P, Q) = \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim Q}[\hat{L}(\boldsymbol{\theta})] + \beta \mathrm{KL}((Q||P)), \tag{3}$$

which is known as the Information Bottleneck (IB) Lagrangian† under the Information Bottleneck Framework Achille & Soatto (2018)Tishby et al. (2000), the Evidence Lower Bound (ELBO) in the variational inference literature Kingma et al. (2015)Bishop (2006) when $\beta = 1$, or more recently as the task complexity Achille et al. (2019). In the above β has the role of regulating the amount of information in the randomized neural network Achille & Soatto (2018), smaller values correspond to more information and potential to overfit.

4 Lower bound on the IB-Lagrangian

The stochastic and non-convex objective 3 is difficult to analyze theoretically. As such we first propose to expand the randomized loss using a Taylor expansion which will make the subsequent analysis tractable. We get

$$\begin{aligned} C_\beta(\mathcal{D}; P, Q) &= \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim Q}[\hat{L}(\boldsymbol{\theta})] + \beta \mathrm{KL}((Q||P)) \\ &\leq \mathop{\mathbb{E}}_{\boldsymbol{\eta} \sim Q'}[\left(\frac{\partial \hat{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\eta}^T \mathbf{H} \boldsymbol{\eta} + O(||\boldsymbol{\eta}||^3)] + \beta \mathrm{KL}((Q||P)) \\ &\approx \mathop{\mathbb{E}}_{\boldsymbol{\eta} \sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H} \boldsymbol{\eta}] + \beta \mathrm{KL}((Q||P)), \end{aligned} \tag{4}$$

where Q' is a centered version of Q. We've made a number of assumptions. In the second line we assumed that the loss $\hat{L}(\boldsymbol{\theta})$ at the minimum is 0. In line 3 we assumed that the gradient $\partial \hat{L}(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ at the minimum is also 0 and the term $O(||\boldsymbol{\eta}||^3)$ is negligible. All assumptions are reasonable for modern deep neural networks. Our subsequent analysis crucially rests on an accurate estimation of the Hessian, which remains an open problem for modern deep learning architectures. Furthermore while we will be minimizing an upper bound on our objective we will be referring with a slight abuse of terminology to our results as a lower bound. Empirically our theoretical predictions are meaningful and should only improve with better estimates of the Hessian.

4.1 Optimal Posterior

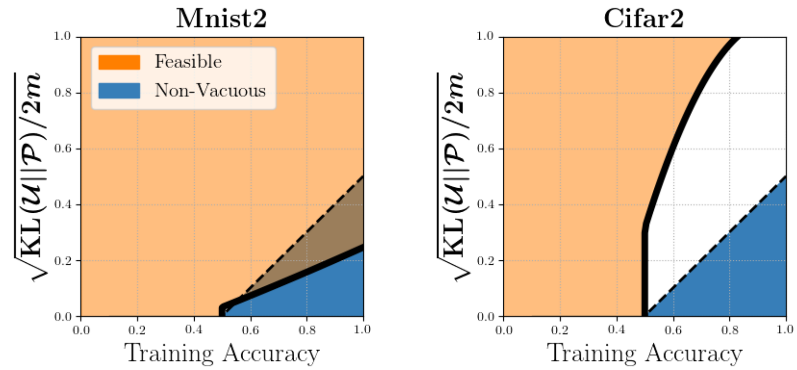
We make the following modeling assumptions $Q = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $P = \mathcal{N}(\boldsymbol{\mu}_1, \lambda \boldsymbol{\Sigma}_1)$ which are popular in VI and PAC-Bayes literature. We can then show that the optimal posterior covariance of the above objective for fixed prior and posterior means has a closed form solution.

Lemma 4.1. *The convex optimization problem* $\min_{\boldsymbol{\Sigma}_0} \mathop{\mathbb{E}}_{\boldsymbol{\eta} \sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_l \boldsymbol{\eta}] + \beta \mathrm{KL}((Q||P))$ *where* $Q = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ *and* $P = \mathcal{N}(\boldsymbol{\mu}_1, \lambda \boldsymbol{\Sigma}_1)$ *has the global minimum:*

$$\boldsymbol{\Sigma}_0^* = \beta(\mathbf{H}_l + \frac{\beta}{\lambda}\boldsymbol{\Sigma}_1^{-1})^{-1}, \tag{5}$$

where $\mathbf{H}_l$ *captures the curvature in the directions of the parameters, while* $\boldsymbol{\Sigma}_1$ *is a chosen prior covariance.*

In practice we perform a grid search over the parameters β and λ and try to find Pareto optimal pairs balancing the accuracy of the randomized classifier and the KL complexity term.



(a)

(b)

Figure 2: **Feasible solutions vs Non-Vacuous solutions**: We merge the 10 classes in Mnist and Cifar to create simpler 2 class problems. For different values of β we compute the optimal complexity terms $\sqrt{\frac{\mathrm{KL}(Q||P)+\ln\frac{2(N-1)}{\delta}}{2N}}$ using 6, 7. We compute the accuracy of the stochastic classifier with MCMC for 5 samples. We plot this lower bound with the solid black line. All points above it are feasible. We see that for the Mnist problem the two regions intersect suggesting that we might be able to prove generalization using Gaussians with diagonal covariance. By contrast in the Cifar case the two regions do not intersect suggesting that the prior and posterior means have a prohibitive distance between them, and we cannot prove generalization with diagonal covariances.

4.2 Optimal Prior

The above solution is not optimal with respect to the *prior* covariance in that we have up to now chosen it arbitrarily. Furthermore given that the choice of the random initialization as the prior mean has been independently shown to result in much tighter bounds in a variety of settings Dziugaite & Roy (2017) Nagarajan & Kolter (2019a) one would wish to isolate the effects of the prior mean on the bound tightness from the prior covariance.

PAC-Bayesian theory allows one to choose an informative prior, however the prior can only depend on the data generating distribution and *not* the training set. A number of previous works Parrado-Hernández et al. (2012)Catoni (2003)Ambroladze et al. (2007) have used this insight mainly on simpler linear settings and usually by training a classifier on a separate training set and using the result as a prior. The concept of a valid prior has been formalized under the differential privacy setting Dziugaite & Roy (2018) where the authors also propose algorithms for the case of deep neural networks.

We ignore these concerns for the moment and optimize the prior covariance directly. The objective is non-convex however for the case of diagonal prior and posterior covariances we manage to find the global minimum.

Theorem 4.2. *The optimal prior and posterior for* $C_\beta(\mathcal{D};P,Q) = \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}_l\boldsymbol{\eta}]+\beta\mathrm{KL}((Q||P))$ *with* $Q = \mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$ *and* $P = \mathcal{N}(\boldsymbol{\mu}_1,\lambda\boldsymbol{\Sigma}_1)$ *and assuming that* $\boldsymbol{\Sigma}_1^{-1} = \boldsymbol{\Lambda}_1 = \mathrm{diag}(\Lambda_{11},\Lambda_{21},...,\Lambda_{k1})$ *and* $\mathbf{H}_l = \mathrm{diag}(h_{1l},h_{2l},...,h_{kl})$ *have:*

$$\Lambda_{i1}^* = \frac{\lambda}{2\beta}[\sqrt{h_{il}^2+\frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}}-h_{il}], \tag{6}$$

$$\Lambda_{i0}^* = \frac{1}{2\beta}[h_{il}+\sqrt{h_{il}^2+\frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}}]. \tag{7}$$

[†] Actually this is an upper bound on the IB Lagrangian, but we will use this term for simplicity.

where $\mathbf{H}_l$ encodes the local curvature at the the minimum, $\boldsymbol{\mu}_1$ corresponds to the random initialization (by design) of the DNN, and $\boldsymbol{\mu}_0$ corresponds to the minimum obtained after optimization.

For our choice of Gaussian prior and posterior, the following is a lower bound on the IB-Lagrangian under any Gaussian prior covariance:

$$\min_{\boldsymbol{\Sigma}_0,\boldsymbol{\Sigma}_1} C_\beta(\mathcal{D};P,Q) \gtrsim \frac{1}{2}(\sum_i a_{il}(\mu_{i0}-\mu_{i1})^2 + \beta\sum_i \ln(\frac{h_{il}+a_{il}}{a_{il}})), \tag{8}$$

where $a_{il} \triangleq a_{il}(\beta,\mu_{i0},\mu_{i1},h_{il}) = \frac{1}{2}[\sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}} - h_{il}]$.

The above result is intuitively pleasing setting a lower bound to what we can achieve which depends only on the initialization (by design), obtained minimum, curvature at the minimum and the regularization parameter β. In particular the scaling factor λ has disappeared.

We now make some important notes about what one can and *cannot* prove using these results, by stressing that the above result is a necessary but not a *sufficient* condition for generalization under our prior and posterior modeling.

- Given a deterministic deep neural network and it's initialization (or other prior mean) one *can* rule out being able to prove generalization using any choice of diagonal covariances when modeling the priors and posteriors as multivariate Gaussians with fixed means. Modeling with other distributions may give different results[†].
- One *cannot* prove generalization using this result, even in the case when the prior mean is valid (only distribution dependent) and the feasible and non-vacuous sets intersect. One still has to compute the prior and posterior covariances in a valid manner. As such a computationally feasible region given finite data and computational resources as well as privacy constraints, might be much smaller than the one we derive here.

We plot our lower bound for simplified 2 class Mnist and Cifar problems in Figure 2. We see that while for the Mnist problem the feasible and non-vacuous regions intersect the same is not true for the Cifar problem. What remains is to see if our results apply for the case of valid priors. First we detail a number of computational issues in section 5.

5 Computational Aspects

We now present a number of computational and memory issues associated with the Hessian of a modern deep neural network. There is ambiguity about the size of the Hessians that can be *computed* exactly Kunstner et al. (2019). There have been few results in this area and the main problem seems to be that the relevant computations are not well supported from common auto-differentiation libraries, such as Tensorflow and Keras. However storing and manipulating the full Hessian of a number of modern deep neural network architectures would be infeasible as the matrix is of size $\mathbf{H} \in \mathbb{R}^{d\times d}$ where d is the number of parameters. As a point of reference a dense uncompressed Numpy matrix for $d = 50000$ takes up 20GB of memory. As such we detail in the next section a number of approximations that make both computing and storing the Hessian feasible.

5.1 Approximating the Full Hessian

As noted in Kunstner et al. (2019) the generalized Gauss-Newton approximation of the Hessian $\mathbf{H}(\boldsymbol{\theta})$ coincides with the Fisher matrix $\mathbf{F}(\boldsymbol{\theta}) = \sum_n \mathbb{E}_{p_{\boldsymbol{\theta}}(y|x_n)}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y|x_n) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y|x_n)^{\mathrm{T}}]$ in modern deep learning architectures. While the Fisher matrix is difficult to compute exactly one can compute an unbiased but noisy estimate as Martens & Grosse (2015)

$$\mathbf{F}(\boldsymbol{\theta}) \approx \sum_n [\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tilde{y}_n|x_n) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tilde{y}_n|x_n)^{\mathrm{T}}], \tag{9}$$

[†] While our result is a formal lower bound on what is achievable by 5, it's applicability on direct minimization of the IB-Lagrangian depends on the tightness of the second order approximation.

where care must be taken to sample $\tilde{y}_n$ from the model predictive distribution $\tilde{y}_n \sim p_{\boldsymbol{\theta}}(y|x_n)$. Additionally we note that the interpretation of the outputs after the softmax as probabilities is not well grounded theoretically Gal & Ghahramani (2016). Determining the true predictive distribution requires MCMC sampling for example by taking multiple dropout samples Gal & Ghahramani (2016).

We now make two additional notes regarding computational aspects of the above. The approximation of the Hessian can be computed efficiently as the outer product of large but manageable gradient vectors. The main computational burden after we approximate the Hessian, and given that we choose a standard normal prior, is inverting a matrix of the form $\tilde{\mathbf{H}} + \alpha I$. This problem can be tackled in a few different ways. The simplest would be to consider only the diagonal elements of $\tilde{\mathbf{H}}$ and the resulting diagonal matrix can be efficiently inverted. However inversion of the full matrix $\tilde{\mathbf{H}} + \alpha I$ is also possible recursively using the Sherman-Morrison formula.

Further issues exist with computing the KL-Divergence of large multivariate Gaussians with non-diagonal covariances in closed form which includes a determinant term that has to be computed with the matrix determinant lemma, as well as sampling efficiently from these distributions. As such we have used the diagonal variant of approximation 9 for our lower bound, but perform a layerwise approximation of the Hessian for all other experiments. We detail this layerwise approximation in the next section, and motivate it theoretically.

5.2 Layerwise approximation

We will now derive an upper bound on 3 which is more suitable for optimization.

Theorem 5.1. *Assuming the following empirical loss $\hat{L}(\boldsymbol{\theta}) = ||f_{\boldsymbol{\theta}}(\mathbf{X}) - \mathbf{Y}||_F$ with $\mathbf{X} = [\boldsymbol{x}_0, ..., \boldsymbol{x}_N]$ and $\mathbf{Y} = [\boldsymbol{y}_0, ..., \boldsymbol{y}_N]$ the following is an upper bound on the IB Lagrangian given that we are at a local minimum:*

$$C_\beta(\mathcal{D}; P, Q) \lesssim \sum_l \sqrt{\sum_j c_{lj} \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'_{lj}} [\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_{lj} \boldsymbol{\eta}]} + \beta \sum_{l,j} \mathrm{KL}((Q_{lj}||P_{lj})), \tag{10}$$

where l denotes different layers, j denotes the different neurons at each layer (we assume the same number for simplicity), $\mathbf{H}_{lj}$ denotes the local Hessian, and Q'_{lj} is a centered version of Q_{lj}. The local Hessian can be computed efficiently as $\mathbf{H}_{lj} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{z}^i_{l-1} {\boldsymbol{z}^i_{l-1}}^T$ and $\boldsymbol{z}^i_{l-1}$ is the latent representation input to layer l for signal i.

We see that we have managed to upper bound the empirical randomized loss by a scaled sum of quadratic terms involving layerwise Hessian matrices and centered random noise vectors. Intuitively we have reduced the complexity of our optimization problem simply by turning it into a number of separate subproblems. The local Hessians can be computed efficiently from outer products of a forward pass of the dataset. Apart from avoiding using backpropagation, breaking the Hessian into subproblems in this manner allows us to move beyond the simplistic diagonal approximation. Implicitly the Hessian now has a block diagonal structure and the blocks are small enough to be inverted directly for the architectures used in this paper. For architectures with larger latent representations the Sherman-Morrison formula can be used instead.

6 Experiments

We now make a number of experiments on the simplified 2 class Mnist and Cifar datasets. Specifically we test the architecture

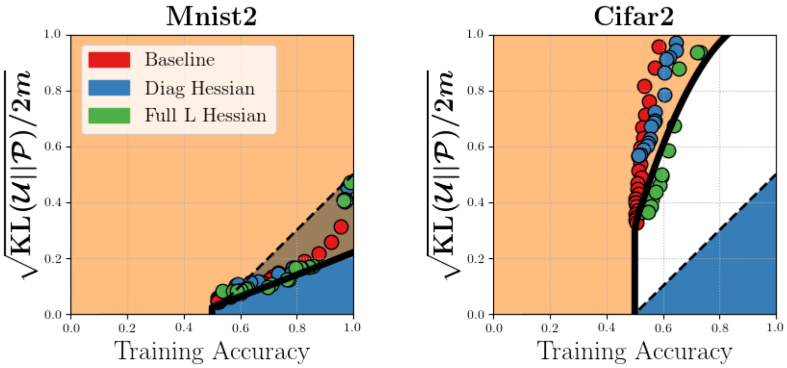
$$\text{input} \rightarrow 300\text{FC} \rightarrow 300\text{FC} \rightarrow \#classes\text{FC} \rightarrow \text{output}$$

on Mnist [*] and

$$\text{input} \rightarrow 200\text{FC} \rightarrow 200\text{FC} \rightarrow \#classes\text{FC} \rightarrow \text{output}$$

on Cifar. We train each configuration to 100% accuracy and derive the layerwise Hessians. We model the prior and posteriors as multivariate Gaussians centered at the initialization and deterministic solution respectively. For the prior we choose the uninformative unit diagonal covariance,

[*] Corresponds to T-300^2 p.7 in Dziugaite & Roy (2017).



(a)

(b)

Figure 3: **Accuracy vs Complexity for different bounds**: We plot $\sqrt{\frac{\mathrm{KL}(Q||P)+\ln\frac{2(N-1)}{\delta}}{2N}}$ and training accuracy (of the randomized classifier) for different architectures and datasets. Points to the right of the dashed line correspond to non-vacuous pairs. All Mnist bounds are non-vacuous. All Cifar bounds are vacuous. We are able to progressively get tighter bounds by using the diagonal Hessian and then the full layerwise Hessian. The improvement is larger over the more difficult Cifar dataset.

scaled by the free parameter λ. The baseline posterior that we use has the same diagonal covariance as the prior. For the baseline we perform a grid search over λ which increases the complexity negligibly. For the optimized posterior we initially test a diagonal approximation of the Hessian "Diag Hessian" which results in an optimal diagonal covariance. We perform a grid search for the parameters λ and β using formula 5 to derive candidates for the optimal posterior covariance. For each point on the grid we calculate the empirical accuracy over the training set using Monte Carlo sampling and 5 samples, as well as the complexity term $\sqrt{\frac{\mathrm{KL}(Q||P)+\ln\frac{2(N-1)}{\delta}}{2N}}$. We then choose the Pareto optimal points from all candidates. We plot the results in Figure 3.

Interestingly we see that for the case of Mnist the baseline is tight with respect to our lower bound and provides non-vacuous bounds. Therefore not much improvement can be achieved using the Hessian approach. This implies a more careful interpretation of the results in Dziugaite & Roy (2017). We see that non-vacuity can also be achieved as a result of the problem being very simple, and the choice of the prior mean being the random initialization. The optimization techniques employed in Dziugaite & Roy (2017) should simply tighten the bound further, mainly by moving the posterior mean closer to the prior mean (the random initialization). For the case of Cifar we see that we can significantly tighten the bound. However we cannot manage to turn a vacuous bound to a non-vacuous one in line with our lower bound. We also test a block-diagonal approximation to the Hessian "Full L Hessian" where each neuron of the network has it's own block. Our non-diagonal layerwise approximation however crude seems to improve significantly over the diagonal case and slightly crosses our diagonal lower bound. This suggests that better approximations of the Hessian as well as better prior means apart from the random initialization might be needed to prove generalization in complex datasets and architectures.

7 Conclusion

We have presented a lower bound on an approximation of the IB-Lagrangian for the case of multivariate Gaussian priors and posteriors with diagonal covariance. This coincides with a lower bound on a PAC-Bayesian generalization bound for an invalid (training set dependent) prior. For cases where the feasible and non-vacuous regions intersect we have seen that it is possible to reach the lower bound and achieve non-vacuous bounds by using valid non-informative priors. We have also presented closed form solutions for the optimal posteriors given fixed means under our modeling assumptions, and motivated theoretically breaking the estimation into layerwise subproblems. Crucially all results depend on high quality estimates of the Hessian which remains an open topic of research for large scale modern deep neural networks.

REFERENCES

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. The information complexity of learning tasks, their structure and their distance. *arXiv preprint arXiv:1904.03292*, 2019.

Amiran Ambroladze, Emilio Parrado-Hernández, and John S Shawe-taylor. Tighter pac-bayes bounds. In *Advances in neural information processing systems*, pp. 9–16, 2007.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Olivier Catoni. A pac-bayesian approach to adaptive classification. *preprint*, 840, 2003.

Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, pp. 4857–4867, 2017.

Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. In *Advances in Neural Information Processing Systems*, pp. 8430–8441, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical fisher approximation. *arXiv preprint arXiv:1905.12558*, 2019.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417, 2015.

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

Vaishnavh Nagarajan and J Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019a.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *arXiv preprint arXiv:1902.04742*, 2019b.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13(Dec):3507–3531, 2012.

Konstantinos Pitas, Andreas Loukas, Mike Davies, and Pierre Vandergheynst. Some limitations of norm based generalization bounds in deep neural networks. *arXiv preprint arXiv:1905.09677*, 2019.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, José Miguel Hernández-Lobato, and Alexander L Gaunt. Deterministic variational inference for robust bayesian neural networks. *arXiv preprint arXiv:1810.03958*, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. *arXiv preprint arXiv:1804.05862*, 2018.

APPENDIX

A. ADDITIONAL EXPERIMENTS

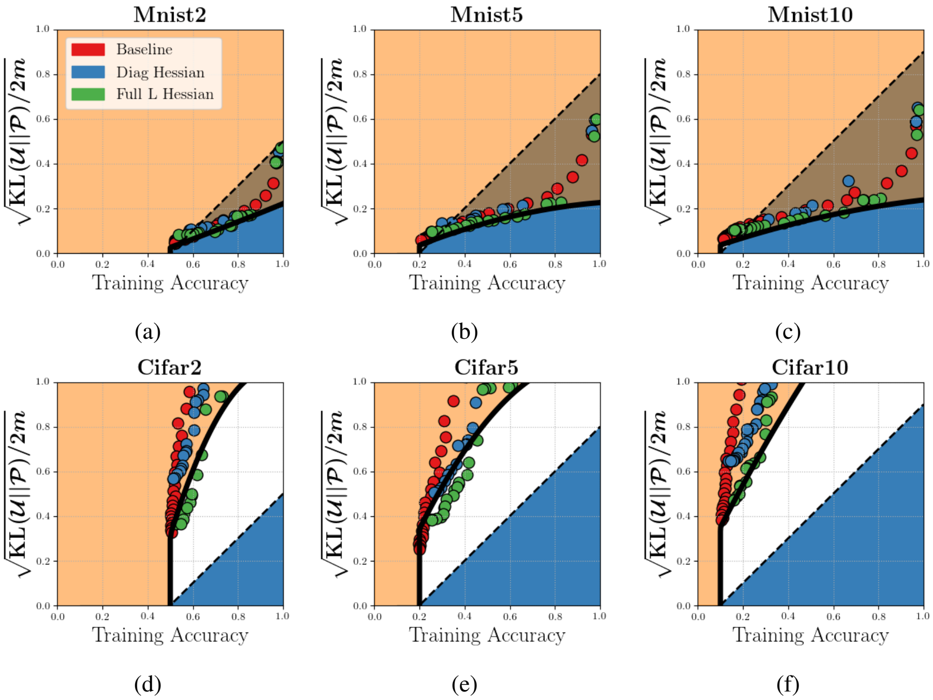
We test the architectures

$$\text{input} \rightarrow 300\text{FC} \rightarrow 300\text{FC} \rightarrow \#classes\text{FC} \rightarrow \text{output}$$

on Mnist and

$$\text{input} \rightarrow 200\text{FC} \rightarrow 200\text{FC} \rightarrow \#classes\text{FC} \rightarrow \text{output}$$

on Cifar. We conduct additional experiments on the original Cifar10 and Mnist10 datasets, as well as Cifar5 and Mnist5 where we merge the 10 classes into 5. The results are consistent across datasets, with more improvement when incorporating the Hessian for more difficult datasets.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: **Accuracy vs Complexity for different bounds**: We plot $\sqrt{\frac{\mathrm{KL}(Q||P)+\ln\frac{2(N-1)}{\delta}}{2N}}$ and training accuracy (of the randomized classifier) for different architectures and datasets. Points to the right of the dashed line correspond to non-vacuous pairs. All Mnist bounds are non-vacuous. All Cifar bounds are vacuous. We are able to progressively get tighter bounds by using the diagonal Hessian and then the full layerwise Hessian. The improvement is larger over the more difficult Cifar dataset.

B. Proofs

Lemma 4.1. The convex optimization problem $\min_{\mathbf{\Sigma}_0} \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_l \boldsymbol{\eta}] + \beta \mathrm{KL}((Q||P))$ where $Q = \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$ and $P = \mathcal{N}(\boldsymbol{\mu}_1, \lambda\mathbf{\Sigma}_1)$ has the global minimum:

$$\mathbf{\Sigma}_0^* = \beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Sigma}_1^{-1})^{-1}, \tag{11}$$

where $\mathbf{H}_l$ captures the curvature in the directions of the parameters, while $\mathbf{\Sigma}_1$ is a chosen prior covariance.

Proof.

$$\begin{aligned} C_\beta(\mathcal{D}; P, Q) = &\mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_l \boldsymbol{\eta}] + \beta \mathrm{KL}((Q||P)) = \\ &\mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\frac{1}{2}\mathrm{tr}(\mathbf{H}_l \boldsymbol{\eta}\boldsymbol{\eta}^T)] + \beta \mathrm{KL}((Q||P)) = \\ &\frac{1}{2}\mathrm{tr}(\mathbf{H}_l \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\boldsymbol{\eta}\boldsymbol{\eta}^T]) + \beta \mathrm{KL}((Q||P)) = \\ &\frac{1}{2}\mathrm{tr}(\mathbf{H}_l \mathbf{\Sigma}_0) + \frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_0) - k + \frac{1}{\lambda}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \\ &+ \ln\left(\frac{\det \lambda\mathbf{\Sigma}_1}{\det \mathbf{\Sigma}_0}\right)) \end{aligned} \tag{12}$$

The gradient with respect to $\mathbf{\Sigma}_0$ is

$$\frac{\partial C_\beta(\mathcal{D}; P, Q)}{\partial \mathbf{\Sigma}_0} = [\frac{1}{2}\mathbf{H}_l + \frac{\beta}{2\lambda}\mathbf{\Sigma}_1^{-1} - \frac{\beta}{2}\mathbf{\Sigma}_0^{-1}]. \tag{13}$$

Setting it to zero, we obtain the minimizer $\mathbf{\Sigma}_0^* = \beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Sigma}_1^{-1})^{-1}$. □

Theorem 4.2. The optimal prior and posterior for $C_\beta(\mathcal{D}; P, Q) = \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_l \boldsymbol{\eta}] + \beta \mathrm{KL}((Q||P))$ with $Q = \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$ and $P = \mathcal{N}(\boldsymbol{\mu}_1, \lambda\mathbf{\Sigma}_1)$ and assuming that $\mathbf{\Sigma}_1^{-1} = \mathbf{\Lambda}_1 = \mathrm{diag}(\Lambda_{11}, \Lambda_{21}, ..., \Lambda_{k1})$ and $\mathbf{H}_l = \mathrm{diag}(h_{1l}, h_{2l}, ..., h_{kl})$ have:

$$\Lambda_{i1}^* = \frac{\lambda}{2\beta}[\sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0} - \mu_{i1})^2}} - h_{il}], \tag{14}$$

$$\Lambda_{i0}^* = \frac{1}{2\beta}[h_{il} + \sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0} - \mu_{i1})^2}}]. \tag{15}$$

where $\mathbf{H}_l$ encodes the local curvature at the the minimum, $\boldsymbol{\mu}_1$ corresponds to the random initialization (by design) of the DNN, and $\boldsymbol{\mu}_0$ corresponds to the minimum obtained after optimization.

For our choice of Gaussian prior and posterior, the following is a lower bound on the IB-Lagrangian under any Gaussian prior covariance:

$$\min_{\mathbf{\Sigma}_0, \mathbf{\Sigma}_1} C_\beta(\mathcal{D}; P, Q) \gtrsim \frac{1}{2}(\sum_i a_{il}(\mu_{i0} - \mu_{i1})^2 + \beta \sum_i \ln(\frac{h_{il} + a_{il}}{a_{il}})), \tag{16}$$

where $a_{il} \triangleq a_{il}(\beta, \mu_{i0}, \mu_{i1}, h_{il}) = \frac{1}{2}[\sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0} - \mu_{i1})^2}} - h_{il}]$.

Proof. Setting $\mathbf{\Lambda}_1 = \mathbf{\Sigma}_1^{-1}$ We can then see that the minimizer is equal to $\mathbf{\Sigma}_0^* = \beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}$. Substituting $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_0^*$ in $C_\beta(\mathcal{D}; P, Q)$ we obtain:

$$\begin{aligned} C_\beta(\mathcal{D}; P, Q)|_{\mathbf{\Sigma}_0=\mathbf{\Sigma}_0^*} =& \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}_l\boldsymbol{\eta}] + \beta\mathrm{KL}((Q||P))|_{\mathbf{\Sigma}_0=\mathbf{\Sigma}_0^*} = \\ & \frac{1}{2}\mathrm{tr}(\mathbf{H}_l\beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}) + \frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\mathbf{\Lambda}_1\beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}) \\ & + \frac{1}{\lambda}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{\Lambda}_1(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) - k + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_1^{-1}}{\det\beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}}\right)) \\ =& \frac{\beta}{2}\mathrm{tr}(\mathbf{H}_l(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}) + \frac{\beta^2}{2\lambda}(\mathrm{tr}(\mathbf{\Lambda}_1(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1})) \\ & + \frac{\beta}{2}(+\frac{1}{\lambda}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{\Lambda}_1(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) - k + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_1^{-1}}{\det\beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}}\right)) \\ =& \frac{\beta}{2}(\mathrm{tr}((\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}) \\ & \frac{1}{\lambda}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{\Lambda}_1(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) - k + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_1^{-1}}{\det\beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}}\right)) \\ =& \frac{\beta}{2}[+\frac{1}{\lambda}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{\Lambda}_1(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_1^{-1}}{\det\beta(\mathbf{H}_l + \frac{\beta}{\lambda}\mathbf{\Lambda}_1)^{-1}}\right)] \end{aligned} \tag{17}$$

The above matrix equation 17 is difficult to deal with directly. We will therefore use the common diagonal approximation of the Hessian which is more amenable to manipulation. Substituting $\mathbf{\Lambda}_1 = \mathrm{diag}(\Lambda_{11}, \Lambda_{21}, ..., \Lambda_{k1})$ and $\mathbf{H}_l = \mathrm{diag}(h_{1l}, h_{2l}, ..., h_{kl})$ in the above expression we get

$$C_\beta(\mathcal{D}; P, Q)|_{\mathbf{\Sigma}_0=\mathbf{\Sigma}_0^*} = \frac{\beta}{2}(\frac{1}{\lambda}\sum_i \Lambda_{i1}(\mu_{i0} - \mu_{i1})^2 - \sum_i \ln(\frac{\Lambda_{i1}}{\lambda}) + \sum_i \ln(\frac{h_{il} + \frac{\beta}{\lambda}\Lambda_{i1}}{\beta})) \tag{18}$$

The above expression is easy to optimize. We see that the sole stationary point exists at

$$\Lambda_{i1}^* = \frac{\lambda}{2\beta}[\sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0} - \mu_{i1})^2}} - h_{il}]. \tag{19}$$

We now turn to the original objective and calculate it's second derivatives. For our diagonal approximation the original objective turns into a sum of separable functions. We will analyze the behavior of one of them for simplicity. The result applies to all other functions in the sum.

$$\begin{aligned} C_\beta(\mathcal{D}; P, Q) =& \sum_i \frac{h_{il}}{2}\nu_{i0} + \sum_i \frac{\beta}{2\lambda}\frac{\nu_{i0}}{\nu_{i1}} - \sum_i \frac{\beta}{2} + \sum_i \frac{\beta(\mu_{i0} - \mu_{i1})^2}{2\lambda}\frac{1}{\nu_{i1}} \\ & + \frac{\beta}{2}[\sum_i \ln(\lambda\nu_{i1}) - \sum_i \ln(\nu_{i0})] \\ =& \sum_i A_i\nu_{i0} + \sum_i B_i\frac{\nu_{i0}}{\nu_{i1}} - \sum_i \frac{\beta}{2} + \sum_i C_i\frac{1}{\nu_{i1}} + D_i[\sum_i \ln(\lambda\nu_{i1}) - \sum_i \ln(\nu_{i0})] \end{aligned} \tag{20}$$

where we have set $A_i = \frac{h_{il}}{2}$, $B_i = \frac{\beta}{2\lambda}$, $C_i = \frac{\beta(\mu_{i0}-\mu_{i1})^2}{2\lambda}$, $D_i = \frac{\beta}{2}$. Denoting $C_{i\beta}(\mathcal{D}; P, Q)$ one function from this sum we calculate

$$\frac{\partial C_{i\beta}(\mathcal{D};P,Q)}{\partial \nu_{i0}} = A_i + \frac{B_i}{\nu_{1i}} - \frac{D_i}{\nu_{i0}}, \quad \frac{\partial C_{i\beta}(\mathcal{D};P,Q)}{\partial \nu_{i1}} = -\frac{B_i \nu_{i0}}{\nu_{i1}^2} - \frac{C_i}{\nu_{i1}^2} + \frac{D_i}{\nu_{i1}} \tag{21}$$

and

$$\frac{\partial C_{i\beta}(\mathcal{D};P,Q)}{\partial^2 \nu_{i0}} = \frac{D_i}{\nu_{i0}^2}, \quad \frac{\partial C_{i\beta}(\mathcal{D};P,Q)}{\partial^2 \nu_{i1}} = 2(B_i\nu_{i0} + C_i)\frac{1}{\nu_{i1}^3} - \frac{D_i}{\nu_{i1}^2} \tag{22}$$

$$\frac{\partial C_{i\beta}(\mathcal{D};P,Q)}{\partial \nu_{i0}\partial \nu_{i1}} = -\frac{B_i}{\nu_{i1}^2}, \quad \frac{\partial C_{i\beta}(\mathcal{D};P,Q)}{\partial \nu_{i1}\partial \nu_{i0}} = -\frac{B_i}{\nu_{i1}^2} \tag{23}$$

We need to check whether the Hessian matrix is PSD so that the stationary point we found is a local minimum and the function is convex. We do that by calculating whether all principal minors of the Hessian are positive.

$$\nabla^2 C_{i\beta}(\nu_{i0}, \nu_{i1}) = \begin{bmatrix} \frac{D_i}{\nu_{i0}^2} & -\frac{B_i}{\nu_{i1}^2} \\ -\frac{B_i}{\nu_{i1}^2} & 2(B_i\nu_{i0} + C_i)\frac{1}{\nu_{i1}^3} - \frac{D_i}{\nu_{i1}^2} \end{bmatrix} \tag{24}$$

We see easily that det$(\frac{D_i}{\nu_{i0}^2}) > 0$. While

$$\begin{aligned} \det(\nabla^2 C_{i\beta}(\nu_{i0}, \nu_{i1})) &= \frac{D_i}{\nu_{i0}^2}\left(2(B_i\nu_{i0} + C_i)\frac{1}{\nu_{i1}^3} - \frac{D_i}{\nu_{i1}^2}\right) - \frac{B_i^2}{\nu_{i1}^4} \\ &= \frac{1}{\nu_{i0}^2\nu_{i1}^4}\left(2C_iD_i\nu_{i1} - (D_i\nu_{i1} - B_i\nu_{i0})^2\right) \\ &= \left(\frac{1}{\nu_{i0}^2\nu_{i1}^4}\frac{\beta^2}{2}\right)\left(\frac{(\mu_{i0} - \mu_{i1})^2}{\lambda}\nu_{i1} - \frac{1}{2}(\nu_{i1} - \frac{\nu_{i0}}{\lambda})^2\right) \end{aligned} \tag{25}$$

A first observation is that this determinant is not always positive and the function is not convex everywhere. However we observe that it is not highly non convex either and the non convexity mainly results from the function tending to infinity logarithmically on one of the boundaries. We now check whether the sole stationary point is always a local minimum. We start by substituting $\nu_{i0}^\star = \beta(h_{il} + \frac{\beta}{\lambda}\frac{1}{\nu_{i1}})^{-1}$ in the multiplicand of 25 as the multiplier is positive by definition

$$\begin{aligned} &\det(\nabla^2 C_{i\beta}(\nu_{i0}^\star, \nu_{i1})) = \frac{1}{{\nu_{i0}^\star}^2\nu_{i1}^4}\frac{\beta^2}{2}\left(\frac{(\mu_{i0} - \mu_{i1})^2}{\lambda}\nu_{i1} - \frac{1}{2}(\nu_{i1} - \frac{\beta}{\lambda}(h_{il} + \frac{\beta}{\lambda}\frac{1}{\nu_{i1}})^{-1})^2\right) \\ &= \frac{1}{{\nu_{i0}^\star}^2\nu_{i1}^4}\frac{\beta^2}{2}\left(\frac{(\mu_{i0} - \mu_{i1})^2}{\lambda}\nu_{i1} - \frac{1}{2}(\nu_{i1} - \frac{\beta}{\lambda}(\frac{\nu_{i1}\lambda}{h_{il}\lambda\nu_{i1} + \beta}))^2\right) \\ &= \frac{1}{{\nu_{i0}^\star}^2\nu_{i1}^4}\frac{\beta^2}{2}\left(\frac{(\mu_{i0} - \mu_{i1})^2}{\lambda}\nu_{i1} - \frac{\nu_{i1}^2}{2}(1 - (\frac{\beta}{h_{il}\lambda\nu_{i1} + \beta}))^2\right) \\ &= \frac{1}{{\nu_{i0}^\star}^2\nu_{i1}^3}\frac{\beta^2}{2}\left(\frac{(\mu_{i0} - \mu_{i1})^2}{\lambda} - \frac{\nu_{i1}}{2}(\frac{h_{il}\lambda\nu_{i1}}{h_{il}\lambda\nu_{i1} + \beta})^2\right) \\ &= \frac{1}{{\nu_{i0}^\star}^2\nu_{i1}^3}\frac{\beta^2}{2}\left(\frac{(\mu_{i0} - \mu_{i1})^2}{\lambda} - \frac{\lambda^2h_{il}^2\nu_{i1}^3}{2(h_{il}\lambda\nu_{i1} + \beta)^2}\right) \\ &= \frac{1}{{\nu_{i0}^\star}^2\nu_{i1}^3 2\lambda(h_{il}\lambda\nu_{i1} + \beta)^2}(2(\mu_{i0} - \mu_{i1})^2(h_{il}\lambda\nu_{i1} + \beta)^2 - \lambda^3h_{il}^2\nu_{i1}^3) \\ &= \frac{1}{{\nu_{i0}^\star}^2 2\lambda(h_{il}\lambda\Lambda_{i1}^{-1} + \beta)^2}(2\Lambda_{i1}(\mu_{i0} - \mu_{i1})^2(h_{il}\lambda + \Lambda_{i1}\beta)^2 - \lambda^3h_{il}^2) \end{aligned} \tag{26}$$

Where we substituted $\nu_{i1} = \Lambda_{i1}^{-1}$ as this will make the calculations easier. We now show a useful identity for $\Lambda_{i1}^\star = \frac{\lambda}{2\beta}[\sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0} - \mu_{i1})^2}} - h_{il}]$

$$
\begin{aligned}
(\Lambda^\star_{i1})^2 &= \frac{\lambda^2}{4\beta^2}\left(h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2} - 2h_{il}\sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}} + h_{il}^2\right) \\
&= \frac{\lambda^2}{4\beta^2}\left(2h_{il}\left(h_{il} - \sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}}\right) + \frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}\right) \\
&= \frac{h_{il}\lambda}{\beta}\frac{\lambda}{2\beta}\left(\left(h_{il} - \sqrt{h_{il}^2 + \frac{4\beta h_{il}}{(\mu_{i0}-\mu_{i1})^2}}\right) + \frac{2\beta}{(\mu_{i0}-\mu_{i1})^2}\right) \\
&= \frac{h_{il}\lambda}{\beta}\left(\frac{\lambda}{(\mu_{i0}-\mu_{i1})^2} - \Lambda^\star_{i1}\right)
\end{aligned}
\tag{27}
$$

We substitute $\Lambda_{i1} = \Lambda^\star_{i1}$ in 26 and again develop only the multiplicand

$$
\begin{aligned}
&\det(\nabla^2 C_{i\beta}(\nu^\star_{i0}, \nu^\star_{i1})) = \frac{1}{{\nu^\star_{i0}}^2 2\lambda(h_{il}\lambda{\Lambda^\star_{i1}}^{-1} + \beta)^2}(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}\lambda + \Lambda^\star_{i1}\beta)^2 - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}\lambda + \Lambda^\star_{i1}\beta)^2 - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + 2h_{il}\lambda\Lambda^\star_{i1}\beta + (\Lambda^\star_{i1})^2\beta^2) - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + 2h_{il}\lambda\Lambda^\star_{i1}\beta + \frac{h_{il}\lambda}{\beta}\left(\frac{\lambda}{(\mu_{i0}-\mu_{i1})^2} - \Lambda^\star_{i1}\right)\beta^2) - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + h_{il}\lambda\Lambda^\star_{i1}\beta + \frac{\beta\lambda^2 h_{il}}{(\mu_{i0}-\mu_{i1})^2}) - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + \frac{\beta\lambda^2 h_{il}}{(\mu_{i0}-\mu_{i1})^2}) + 2(\Lambda^\star_{i1})^2(\mu_{i0}-\mu_{i1})^2 h_{il}\lambda\beta - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + \frac{\beta\lambda^2 h_{il}}{(\mu_{i0}-\mu_{i1})^2}) \\
&+ 2\frac{h_{il}\lambda}{\beta}\left(\frac{\lambda}{(\mu_{i0}-\mu_{i1})^2} - \Lambda^\star_{i1}\right)(\mu_{i0}-\mu_{i1})^2 h_{il}\lambda\beta - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + \frac{\beta\lambda^2 h_{il}}{(\mu_{i0}-\mu_{i1})^2}) + 2\lambda^3 h_{il}^2 - 2h_{il}^2\lambda^2(\mu_{i0}-\mu_{i1})^2\Lambda^\star_{i1} - \lambda^3 h_{il}^2) \\
&= A_i(2\Lambda^\star_{i1}(\mu_{i0}-\mu_{i1})^2(h_{il}^2\lambda^2 + \frac{\beta\lambda^2 h_{il}}{(\mu_{i0}-\mu_{i1})^2}) + \lambda^3 h_{il}^2 - 2h_{il}^2\lambda^2(\mu_{i0}-\mu_{i1})^2\Lambda^\star_{i1}) \\
&= A_i(2\Lambda^\star_{i1}\beta\lambda^2 h_{il} + \lambda^3 h_{il}^2) \\
&> 0
\end{aligned}
\tag{28}
$$

where we have set $A_i = \frac{1}{{\nu^\star_{i0}}^2 2\lambda(h_{il}\lambda(\Lambda^\star_{i1})^{-1} + \beta)^2} > 0$. We have used 27 in lines 4 and 7.

Indeed the stationary point is always a local minimum. What remains is to show that there are no other local minima at the boundaries of the domain. From 20 we see that we only need to evaluate expressions of the form $f(\nu_{i0}) = \nu_{i0} - \ln(\nu_{i0})$ and $g(\nu_{i1}) = \frac{1}{\nu_{i0}} + \ln(\nu_{i0})$. By application of L'Hôpital's rule it's easy to show that

$$
\begin{aligned}
\lim_{\substack{\nu_{i0}\to 0\\ \nu_{i1}=\text{ct}}} C_{i\beta}(\nu_{i0}, \nu_{i1}) &= \lim_{\substack{\nu_{i0}\to +\infty\\ \nu_{i1}=\text{ct}}} C_{i\beta}(\nu_{i0}, \nu_{i1}) \\
&= \lim_{\substack{\nu_{i0}=\text{ct}\\ \nu_{i1}\to 0}} C_{i\beta}(\nu_{i0}, \nu_{i1}) = \lim_{\substack{\nu_{i0}=\text{ct}\\ \nu_{i1}\to +\infty}} C_{i\beta}(\nu_{i0}, \nu_{i1}) = +\infty
\end{aligned}
\tag{29}
$$

this concludes the proof.

□

Theorem 5.1. Assuming the following empirical loss $\hat{L}(\boldsymbol{\theta}) = ||f_{\boldsymbol{\theta}}(\mathbf{X}) - \mathbf{Y}||_F$ with $\mathbf{X} = [\boldsymbol{x}_0, ..., \boldsymbol{x}_N]$ and $\mathbf{Y} = [\boldsymbol{y}_0, ..., \boldsymbol{y}_N]$ the following is an upper bound on the IB Lagrangian given that we are at a local minimum:

$$C_\beta(\mathcal{D}; P, Q) \lesssim \sum_l \sqrt{\sum_j c_{lj} \mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'_{lj}}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_{lj}\boldsymbol{\eta}]} + \beta \sum_{l,j} \mathrm{KL}((Q_{lj}||P_{lj})), \tag{30}$$

where l denotes different layers, j denotes the different neurons at each layer (we assume the same number for simplicity), $\mathbf{H}_{lj}$ denotes the local Hessian, and Q'_{lj} is a centered version of Q_{lj}. The local Hessian can be computed efficiently as $\mathbf{H}_{lj} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{z}_{l-1}^i {\boldsymbol{z}_{l-1}^i}^T$ and $\boldsymbol{z}_{l-1}^i$ is the latent representation input to layer l for signal i.

Proof. We start by defining a layerwise empirical error $\hat{E}_l(\boldsymbol{\theta}_l) := \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{W}_l \boldsymbol{z}_{l-1}^i - \boldsymbol{z}_l^i||_2^2$. One can then easily show that $\hat{L}(\boldsymbol{\theta}) \leq \sum_{k=1}^{L-1} \sqrt{\hat{E}_l(\boldsymbol{\theta}_l)} \prod_{k=l+1}^{L} ||\hat{\boldsymbol{\theta}}_k||_F + \sqrt{\hat{E}_L(\boldsymbol{\theta}_L)}$ Dong et al. (2017) substituting this in the IB Lagrangian we get

$$\begin{aligned}
C_\beta(\mathcal{D}; P, Q) &= \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{L}(\boldsymbol{\theta})] + \beta \mathrm{KL}((Q||P)) \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\sum_{l=1}^{L-1} \sqrt{\hat{E}_l(\boldsymbol{\theta}_l)} \prod_{k=l+1}^{L} ||\hat{\boldsymbol{\theta}}_k||_F + \sqrt{\hat{E}_L(\boldsymbol{\theta}_L)}] + \beta \mathrm{KL}((Q||P)) \\
&\leq \sum_{l=1}^{L-1} \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{E}_l(\boldsymbol{\theta}_l)]} \prod_{l=k+1}^{L} \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[||\hat{\boldsymbol{\theta}}_l||_F] + \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{E}_L(\boldsymbol{\theta}_L)]} + \beta \mathrm{KL}((Q||P)) \\
&\leq \sum_{l=1}^{L} c_l \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim Q}[\hat{E}_l(\boldsymbol{\theta}_l)]} + \beta \mathrm{KL}((Q||P)) \\
&\leq \sum_{l=1}^{L} c_l \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\left(\frac{\partial \hat{E}_l(\boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l}\right)^T \boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_l \boldsymbol{\eta} + O(||\boldsymbol{\eta}||^3)]} + \beta \mathrm{KL}((Q||P)) \\
&\approx \sum_{l=1}^{L} c_l \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\eta}\sim Q'}[\frac{1}{2}\boldsymbol{\eta}^T \mathbf{H}_l \boldsymbol{\eta}]} + \beta \mathrm{KL}((Q||P))
\end{aligned} \tag{31}$$

were in line 3 we use the linearity of expectation, Hölder's inequality due to the non-negativity of the random variables, and Jensen's inequality for the concave square root. In line 4 we hide the Frobenius terms into constants c_l. Each error term $\hat{E}_l(\boldsymbol{\theta}_l)$ is only multiplied with Frobenius norm terms $||\hat{\boldsymbol{\theta}}_l||_F$ from the deeper layers. Therefore one can start optimizing from the final layer and proceed to the first while considering c_l as constant. In practice we will just consider all c_l as unknown scaling factors. In line 5 we expand each $\hat{E}_l(\boldsymbol{\theta}_l)$ term using a Taylor expansion, and subsequently ignore the first term as the DNN is assumed to be well trained and the first derivative will be zero, while terms with order higher than 2 are unimportant. We also use Q' to denote the centered version of distribution Q.

Taking the first and second derivatives of the layerwise error with respect to $\mathbf{W}_l$ we get

$$\frac{\partial E_l(\boldsymbol{\theta})}{\partial \mathbf{W}_l} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial}{\partial \mathbf{W}_l} ||\mathbf{W}_l \boldsymbol{z}_{l-1}^i - \boldsymbol{z}_l^i||_2^2 = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{W}_l \boldsymbol{z}_{l-1}^i - \boldsymbol{z}_l^i) 2 {\boldsymbol{z}_{l-1}^i}^T \tag{32}$$

$$\frac{\partial^2 E_l(\boldsymbol{\theta})}{\partial \mathbf{W}_l \partial \mathbf{W}_l^{(j,:)}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{z}_{l-1}^i {\boldsymbol{z}_{l-1}^i}^T \tag{33}$$

Where the second derivative is with respect to any row $\mathbf{W}_l^{(j,:)}$ of the weight matrix $\mathbf{W}_l$. We see that the full Hessian matrix $\mathbf{H}_l = \frac{\partial^2 E_l(\boldsymbol{\theta})}{\partial^2 \mathbf{W}_l}$ then has a block diagonal structure where each block is equal to $\mathbf{H}_{lj} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{z}_{l-1}^i {\boldsymbol{z}_{l-1}^i}^T$. Each row $\mathbf{W}_l^{(j,:)}$ corresponds to a neuron of the layer and for an appropriate choice of prior and posterior with block diagonal covariances it is easy to see that the final form of expression 31 factorizes as

$$C_\beta(\mathcal{D}; P, Q) \lesssim \sum_l \sqrt{\sum_j c_{lj} \mathop{\mathbb{E}}_{\boldsymbol{\eta} \sim {Q'}_{lj}} [\frac{1}{2} \boldsymbol{\eta}^T \mathbf{H}_{lj} \boldsymbol{\eta}]} + \beta \sum_{l,j} \mathrm{KL}((Q_{lj} || P_{lj})) \tag{34}$$

this completes the proof. □