Hierarchical PAC-Bayes Bounds via Deep Probabilistic Programming

Jonathan Warrell^{1,2} **and Mark Gerstein**^{1,2,3}
^{1} Program in Computational Biology and Bioinformatics, Yale University
^{2} Molecular Biophysics and Biochemistry, Yale University
^{3} Department of Computer Science, Yale University
`jonathan.warrell@yale.edu`, `mark@gersteinlab.org`

Abstract

PAC-Bayes approaches have recently generated some of the tightest generalization bounds for neural networks, as well as providing objective functions for regularization when training networks *de novo*, and in the context of transfer learning. However, existing approaches often place restrictions on the form of the prior and/or posterior. We show how general and tractable PAC-Bayes bounds can be derived in a deep probabilistic programming (DPP) framework. This allows both prior and posterior to be arbitrary DPPs, hyper-priors to be easily introduced, and variational techniques to be used during optimization. We test our framework using generalization and transfer learning tasks on synthetic and biological data.

1 Introduction

Some of the tightest recent generalization bounds for neural networks have used PAC-Bayes approaches [1-3]. This success has depended in part on using *data dependent* priors; while the prior in PAC-Bayes cannot explicitly depend on the observed dataset, it may depend on the generating data distribution. To achieve this, approaches have used priors which depend on a subset of the training data [4,5], are implicitly defined [5,6], or use privacy-preserving methods [1,2]. Other approaches have introduced a hyper-prior and hyper-posterior in the context of transfer learning [7,8], where the priors for individual tasks may be sampled from a data-dependent hyper-posterior. Such bounds may also be used as regularizers for *de novo* training [5], and for transfer learning [7,8].

PAC-Bayes bounds are formulated in terms of prior and posterior stochastic classifiers, P and Q. The expected risk of Q is bounded by the empirical Gibbs risk and a term involving $\text{KL}(Q, P)$. Both P and Q may naturally be treated as deep probabilistic programs (DPPs, [9-11]); from this perspective, they are simply stochastic functions. An advantage of this perspective is that it places minimal restrictions on their functional forms; existing approaches such as above focus on restricted forms of the prior [1-8] and/or posterior [4-6]. Further, hyper-priors and -posteriors may be treated as higher-order stochastic programs which return stochastic classifiers, and recent variational methods [10,11] used for efficient optimization. Here, we show how general and tractable hierarchical PAC-Bayes bounds can be derived in a DPP framework, developed in a stochastic type system adapted from [12]. We use recent variational techniques [13-15] to develop modified objectives from existing bounds, which simultaneously serve as valid (looser) generalization bounds. We test our framework using generalization and transfer learning tasks on synthetic and biological data.

2 PAC-Bayes Bounds as Deep Probabilistic Programs

Probabilistic type system. We assume we have types $A, B, C..., Z$ along with function types (e.g. $A \rightarrow B$), and write $a : A$ for a belongs to type A. The type I denotes the unit interval, and we write $A' = (A \rightarrow I)$ for the type of distributions over A, where we assume for convenience all types are discrete, and A' contains only maps which sum to 1. For $\pi : A'$, we use the special notation $(\text{samp } \pi)$ to denote a sampling procedure (probabilistic program) which draws from π. The term $(\text{samp } \pi)$ may be reduced probabilistically by ρ-reduction [12] by sampling; a sample so drawn is denoted π^*, and hence $\pi^* : A$. We can compose sampling procedures; hence, if we have $f : A^2 \rightarrow B$

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

and $\pi_1, \pi_2 : A'$, we may form the term $t = f(\text{samp } \pi_1, \text{samp } \pi_2)$. We assign sampling procedures to the type of distributions they implicitly represent; hence $t : B'$ (unlike [12], which leaves such terms untyped). Further, we may assign multiple levels to the sampling statements within a term. For this purpose, we use the notation $t^+, t^{++}, ...,$ for $(\text{samp}_1\ t), (\text{samp}_2\ t), ...$ Here, ρ-reduction reduces only the first level samp statements in a term, and decrements by one the levels of all others; hence if we annotate t above as $t_1 = f(\pi_1^+, \pi_2^{++})$, this reduces by ρ-reduction to $t_1^* = f(\pi_1^*, \pi_2^+)$, and we have the type assignment $t_1 : B''$. As noted, a sampling procedure implicitly represents a distribution. Hence, for $a_1 = (\text{samp } \pi) : A'$ we have $a_1(a_0) = P(a_1^* = a_0)$ which we will also write as $P_{a_1}(a_0)$. With this notation, we can then define the KL-divergence between two probabilistic programs. Letting $p, q : X'$, we set $\text{KL}(q, p) = \sum_{x:X} P_q(x) \log(P_q(x)/P_p(x))$. Note that this treats explicit and implicitly defined distributions identically. The type system defined above is the minimal system for our purposes; λ-terms and dependent types [12] may also be introduced for a more powerful system.

Stochastic classifier models. We next state explicitly the stochastic classifier formulations we use to define priors, posteriors, and hyper-priors and -posteriors in a PAC-Bayes setting. Here, assume we have input and output types X and Y. Further, let Z represent fixed-precision positive and negative reals. We use the fixed notation $\text{N}(.; \mu, \Sigma)$ to represent a multivariate normal (belonging to type $Z^n \rightarrow I$), and $\text{NN}_{T_1,T_2}(.; \theta)$ to represent a neural network with parameters θ (belonging to function type $T_1 \rightarrow T_2$ for some types T_1, T_2). We then define a hierarchy of types: $F_0 = (X \rightarrow Y)$, $F_1 = F_0' = (X \rightarrow Y) \rightarrow I$, $F_2 = F_0''$, and so on. Here, F_0 is the type of deterministic classifiers (or regression models) between X and Y; F_1 represents distributions over F_0, corresponding to stochastic classifiers; and F_2 represents distributions over F_1, forming a type in which hyper-priors and -posteriors are represented. We can specify flexible models at all these levels via the following probabilistic programs, $f_0 : F_0$, $f_1 : F_1$, $f_2 : F_2$:

$$\begin{aligned} f_0 &= \text{NN}_{X,Y}(.; \theta_0) \\ f_1 &= \text{NN}_{X,Y}(.; \text{NN}_{Z^S,\Theta_0}(z_1^+; \theta_1) + e_1^+) \\ f_2 &= \text{NN}_{X,Y}(.; \text{NN}_{Z^S,\Theta_0}(z_1^{++}; \text{NN}_{Z^S,\Theta_1}(z_2^+, \theta_2) + e_2^+) + e_1^{++}) \end{aligned} \tag{1}$$

Here, Θ_0, Θ_1 are the parameter spaces (types) for θ_0, θ_1, $z_1, z_2 = \text{N}(.; \mathbf{0}_S, \mathbf{I}_S)$ are standard normal latent variables (where S is the dimensionality of the latent space), and $e_1 = \text{N}(.; \mathbf{0}_{|\Theta_0|}, \sigma\mathbf{I}_{|\Theta_0|})$ is a noise term (similarly for e_2, substituting Θ_1 for Θ_0). We note that f_0, f_1, f_2 are entirely specified by the parameter vectors $\theta_0, \theta_1, \theta_2$ respectively. Hence, the types F_0, F_1, F_2 are isomorphic to $\Theta_0, \Theta_1, \Theta_2$ (if terms are restricted to the forms in Eq. 1), and for two such programs $f_1^a, f_1^b : F_1$, $\text{KL}(f_1^a, f_1^b)$ can be estimated by approximating an integral across Θ_0.

Hierarchical PAC-Bayes bounds. We state below two objective functions derived from PAC-Bayes bounds, using the notation above, derived from [16] and [8] respectively:

$$\phi^1(f_1^\rho) = R(f_1^\rho) + (1/\lambda)[\text{KL}(f_1^\rho, f_1^\pi) + \log(1/\delta) + (\lambda^2/M)] \tag{2}$$

$$\begin{aligned} \phi^2(f_2^\rho, f_1^{\rho,1}, f_1^{\rho,2}...f_1^{\rho,N}) =&< R(f_1^{\rho,n}) > + < ((\text{KL}(f_2^\rho, f_2^\pi) + \text{KL}(f_1^{\rho,n}, (f_2^\pi)^*) + a)/b)^{1/2} > \\ &+ ((\text{KL}(f_2^\rho, f_2^\pi) + c)/d)^{1/2} \end{aligned} \tag{3}$$

Here, f_1^π, f_2^π denote priors and hyper-priors respectively, and f_1^ρ, f_2^ρ posteriors and hyper-posteriors. R is the empirical (Gibbs) risk, M, N and M_n are the number of training examples, tasks, and examples for task n respectively, $< . >$ denotes the average as n ranges over tasks, $a = \log(2NM_n/\delta)$, $b = 2(M_n - 1)$, $c = \log(2N/\delta)$, $d = 2(N-1)$, and λ, δ are hyper-parameters. A difficulty in optimizing these bounds directly is deriving estimators for the KL terms which are unbiased, or preserve the upper-bound. In Appendix A, we outline three approaches, deriving a series of modified PAC-bounds for tractable optimization. The first (Eq. 4, used in our experimentation) splits the KL terms into entropy and cross-entropy components, $\text{KL}(q, p) = -H(q) + H(q, p)$. Here, an upper-bound may be used on the negative-entropy term, for instance $-H(q) \leq E_{q(x,\gamma)}[\log q(\gamma) - \log q(x|\gamma) + \log r(\gamma|x)]$, as introduced in [14], where (in our terms) q is a probabilistic program, γ ranges across the latent space, and r is an auxiliary variational distribution. The cross-entropy $H(q, p)$ may be approximated by Monte-Carlo, or a variational (ELBO [17]) bound, both giving upper-bounds on the term. The second approach (Eq. 5) uses a direct upper-bound on the KL-divergence [15]. The third (Eq. 6) splits the KL term as above, but uses the CUBO bound [13] to bound the entropy term. Further, by using the models in Eq. 1 in all three approaches, a scaled ELBO-bound may be substituted for the risk

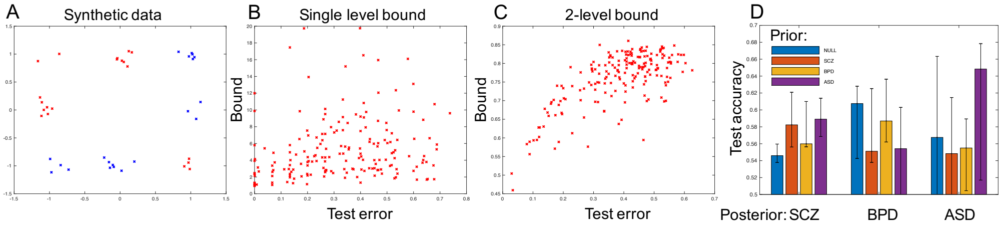

Figure 1: Testing generalization and transfer learning. (A) shows example synthetic data, while (B) and (C) compare the test error and generalization bounds achieved by optimizing Eqs. 2 and 3 on synthetic data. (D) shows results for transfer learning on genomics data, where prior and posterior are trained to identify different psychiatric conditions vs controls (Eq. 2). Error bars show quartiles.

term using the reparameterization trick, which bounds the Gibbs risk. We give explicit forms for all bounds in App. A; the first two are valid generalization bounds, while the third is only approximate if the CUBO is estimated via Monte-Carlo [13]. We also discuss the use of amortization [10,11].

3 Results

Synthetic experiments. We test the ability of single and multi-level DPP-based bounds to predict generalization on synthetic data. For this purpose, we design a synthetic set, having 33 tasks, each being a binary classification problem with 2d input features, where the inputs fall into 8 clusters arranged as shown in Fig. 1A, with 4 being randomly assigned to classes 0 and 1 on each task. This allows for transfer of information across tasks, since similar decision boundaries may occur in multiple tasks. For each task, we generate 6 datasets with varying levels of noise added (to permit different levels of generalization), flipping 0, 20, 40, 60, 80 and 100% of the labels, and split the data into training, validation and testing partitions of $N = 15$ data-points each. We first learn a stochastic classifier f_1^ρ using Eq. 4 on the validation partition, after pre-training a prior f_1^π on the training partition using the ELBO bound [17]. Fig. 1B plots the test error against the bound, which are significantly correlated ($r = 0.2, p = 0.008$). Further, a regression of the test error on the training error and bound show the bound to be moderately informative ($p = 0.1$, 1-tailed ANOVA). We then use a multi-level bound modified from Eq. 3 (see App. A) to learn classifiers f_1^ρ for each task, while simultaneously fitting a hyper-posterior f_2^ρ to groups of 3 tasks at a time (using the validation sets only). Fig. 1C shows this approach is able to achieve a better correlation between the bound and test error ($r = 0.7, p = 2e-30$), and that the bound carries significant additional information about the test error versus the training error alone ($p = 0.01$, 1-tailed ANOVA), showing that the hierarchical approach is able to share information between tasks. We compare against the model of [8], in which the priors, and hyper-posterior/prior are restricted to be Gaussian in form, which achieves significantly lower test performance across tasks ($p = 0.015$, 1-tailed t-test, 0.53 vs 0.56 mean accuracy), showing the flexibility afforded by the DPP formulation to be beneficial. In all cases, we use networks with 2 hidden layers of 5 units each, a 2-d latent space, set $\sigma = 0.1, \lambda = 10, \delta = 0.05$, and use Eq. 4 and its 2-level analogue for optimization.

Psychiatric genomics data. We further test our approach on psychiatric genomics data from the PsychENCODE project [18], consisting of gene expression (RNA-Seq) levels from post-mortem prefrontal cortex samples of control, schizophrenia (SCZ), bipolar (BDP) and autistic (ASD) subjects. We create datasets balanced for cases and controls (and covariates, see [18]) for each disorder, with 710, 188 and 62 subjects respectively, from which we create 10 training, validation and testing partitions (approx. 0.45/0.45/0.1 split). We then replicate the setting of the first synthetic experiment above, training priors f_1^π on each of the training partitions (via an ELBO objective), before training a posterior stochastic classifier f_1^ρ using Eq. 4 on the validation data; further, we test all combinations of disorders when learning priors and posteriors. The results in Fig. 1D show that both SCZ and ASD models are able to use the information in the prior to improve generalization. The SCZ results are particularly interesting, in that the priors trained on all 3 disorders are able to improve the baseline model; the improvements for the SCZ and BPD priors here are significant ($p = 0.006$ and $p = 0.026$ respectively, 1-tailed t-test). In the ASD case, only the ASD prior improves performance, while for BPD, no improvement is gained. We note that the SCZ dataset is substantially larger than the other disorders', which may effect the results. In general, the SCZ results point to a shared etiology of psychiatric conditions, as has been highlighted recently [19,20]. Further, we compare against a model in which the prior is restricted to be Gaussian in form as in [5], observing significantly lower performance across models ($p = 9.9e-3$, 1-tailed t-test, 0.57 vs 0.59 mean accuracy). For each data split, we select the 5 most discriminative genes for each disorder using the training partitions to create a 15-d input space; the network hyper-parameters and bounds are identical to the synthetic case.

4 Discussion

We have shown how hierarchical PAC-Bayes bounds can be naturally converted into training objectives in a probabilistic programming framework, and have proposed modified forms of these bounds which can be readily optimized using existing variational methods, while continuing to serve as valid generalization bounds. Through experiments on synthetic and biological data, we have shown the potential of these objectives to predict generalization and perform transfer learning. A natural future direction is the extension of the hierarchical objective in Eq. 3 to higher-order priors and posteriors (which are readily formulated by extending Eq. 1). It may be possible to extend the framework of [8] to generate valid high-order generalization bounds for this purpose; however, we note that new techniques may be required to tighten these bounds to be non-vacuous (although loose bounds may be function as useful learning objectives and carry information about generalization, as our results show).

References

[1] Dziugaite, G. K., & Roy, D. M. (2018). Data-dependent PAC-Bayes priors via differential privacy. In *Advances in Neural Information Processing Systems* (pp. 8430-8441).

[2] Dziugaite, G.K. and Roy, D.M., 2017. Entropy-SGD optimizes the prior of a PAC-Bayes bound: Generalization properties of Entropy-SGD and data-dependent priors. arXiv preprint arXiv:1712.09376.

[3] Zhou, W., Veitch, V., Austern, M., Adams, R. P., & Orbanz, P. (2018). Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. arXiv preprint arXiv:1804.05862.

[4] Ambroladze, A., Parrado-Hernández, E., & Shawe-taylor, J. S. (2007). Tighter pac-bayes bounds. In *Advances in neural information processing systems* (pp. 9-16).

[5] Parrado-Hernández, E., Ambroladze, A., Shawe-Taylor, J., & Sun, S. (2012). PAC-Bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13(Dec), 3507-3531.

[6] Rivasplata, O., Szepesvari, C., Shawe-Taylor, J. S., Parrado-Hernandez, E., & Sun, S. (2018). PAC-Bayes bounds for stable algorithms with instance-dependent priors. In *Advances in Neural Information Processing Systems* (pp. 9214-9224).

[7] Pentina, A. and Lampert, C., 2014. A PAC-Bayesian bound for lifelong learning. In *International Conference on Machine Learning* (pp. 991-999).

[8] Amit, R. and Meir, R., 2017. Meta-learning by adjusting priors based on extended PAC-Bayes theory. arXiv preprint arXiv:1711.01244.

[9] Goodman, N., Mansinghka, V., Roy, D. M., Bonawitz, K., & Tenenbaum, J. B. (2012). Church: a language for generative models. arXiv preprint arXiv:1206.3255.

[10] Tran, D., Hoffman, M. D., Saurous, R. A., Brevdo, E., Murphy, K., & Blei, D. M. (2017). Deep probabilistic programming. arXiv preprint arXiv:1701.03757.

[11] Tran, D., Hoffman, M. W., Moore, D., Suter, C., Vasudevan, S., & Radul, A. (2018). Simple, distributed, and accelerated probabilistic programming. In *Advances in Neural Information Processing Systems* (pp. 7598-7609).

[12] Warrell J., & Gerstein M. (2018) Dependent Type Networks: A Probabilistic Logic via the Curry-Howard Correspondence in a System of Probabilistic Dependent Types. In *Uncertainty in Artificial Intelligence, Workshop on Uncertainty in Deep Learning*. http://www.gatsby.ucl.ac.uk/~balaji/udl-camera-ready/UDL-19.pdf

[13] Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., & Blei, D. (2017). Variational Inference via χ Upper Bound Minimization. In *Advances in Neural Information Processing Systems* (pp. 2732-2741).

[14] Ranganath, R., Tran, D., & Blei, D. (2016, June). Hierarchical variational models. In *International Conference on Machine Learning* (pp. 324-333).

[15] Sobolev, A. and Vetrov, D., 2019. Importance Weighted Hierarchical Variational Inference. arXiv preprint arXiv:1905.03290.

[16] Alquier, P., Ridgway, J., & Chopin, N. (2016). On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 17(1), 8374-8414.

[17] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[18] Wang, D., Liu, S., Warrell, J., Won, H., Shi, X., Navarro, F. C., ... & Gerstein, M. (2018). Comprehensive functional genomic resource and integrative model for the human brain. *Science*, 362(6420), eaat8464.

[19] Anttila, V., Bulik-Sullivan, B., Finucane, H. K., Walters, R. K., Bras, J., Duncan, L., ... & Neale, B. (2018). Analysis of shared heritability in common disorders of the brain. *Science*, 360(6395), eaap8757.

[20] Gandal, M. J., Haney, J. R., Parikshak, N. N., Leppa, V., Ramaswami, G., Hartl, C., ... & Geschwind, D. (2018). Shared molecular neuropathology across major psychiatric disorders parallels polygenic overlap. *Science*, 359(6376), 693-697.

Appendix A: Modified PAC-Bayes Bounds

We state here the explicit forms of the modified bounds which may be used as objectives in training the models from Eq. 1. For convenience, we give these modified versions in the context of the single-level bound (Eq. 2). For the first modified bound (which we use in our experimentation), we substitute an ELBO bound for the empirical risk, and use an upper-bound on the negative entropy $(-H(q(x)) \leq E_{q(x,\gamma)}[\log q(\gamma) - \log q(x|\gamma) + \log r(\gamma|x)]$, as introduced in the context of hierarchical variational models in [14]):

$$\begin{aligned}\phi_a^1(f_1^\rho, r_1, r_2) \quad = \quad & -E_{r_1(\gamma|x,y)}[C\log(f_1^\rho(y|x,\gamma)] + C\cdot \mathrm{KL}(r_1(\gamma|x,y), z_1) + \\ & (1/\lambda)[E_{z_1(\gamma)f_1^\rho(\theta_0|\gamma)}[\log z_1(\gamma) - \log f_1^\rho(\theta_0|\gamma) + \log r_2(\gamma|\theta_0)] - \\ & E_{f_1^\rho}[\log(f_1^\pi(\theta_0))] + \log(1/\delta) + (\lambda^2/M)] \qquad (4)\end{aligned}$$

where $C = 1/\log 2$ (scaling the ELBO to bound the Gibbs risk), and r_1, r_2 are variational distributions. Here, r_1 has the type $(X,Y) \rightarrow (Z^S)'$; hence it maps pairs of inputs/outputs to distributions over the latent space, and $\gamma : Z^S$. By contrast, r_2 has type $\Theta_0 \rightarrow (Z^S)'$. Further, we note the slight abuse of notation: $f_1^\rho(y|x,\gamma) = P((\text{samp } f_1^\rho)(x) = y)$. Finally, as discussed, since Eq. 4 upper-bounds Eq. 2, it remains an upper-bound on the expected risk (i.e. expected test error).

An alternative to Eq. 4 is to directly upper-bound the KL divergence term as in [15]. We give this modified bound below, stated in terms of the empirical risk for simplicity:

$$\begin{aligned}\phi_b^1(f_1^\rho, r_1, r_2) \quad = \quad & R(f_1^\rho) + (1/\lambda)[E_{f_1^\rho(\theta_0|\gamma_0)z_1(\gamma_0)}E_{r_1(\gamma^a_{1:K}|\theta_0)}E_{r_2(\gamma^b_{1:L}|\theta_0)}\log F_{\theta_0,\gamma_0,\gamma^a_{1:K},\gamma^b_{1:L}} + \\ & + \log(1/\delta) + (\lambda^2/M)] \qquad (5)\end{aligned}$$

where $F_{\theta_0,\gamma_0,\gamma^a_{1:K},\gamma^b_{1:L}} = (A/B)$, for $A = (1/(1+K)\sum_{k=0:K}(f_1^\rho(\theta_0|\gamma^a_k)z_1(\gamma_k))/(r_1(\gamma^a_k|\theta_0))$ and $B = (1/(1+L)\sum_{l=1:L}(f_1^\pi(\theta_0|\gamma^b_l)z_1(\gamma_l))/(r_2(\gamma^b_l|\theta_0))$. Here, r_1, r_2 both have type $\Theta_0 \rightarrow (Z^S)'$, and the bound has a multisample form, with the γ's all being samples in the latent space Z^S. Again, Eq. 5 upper-bounds Eq. 2, and so upper-bounds the expected risk.

A further alternative is to use the CUBO bound [13] to upper-bound the negative entropy term:

$$\begin{aligned}\phi_c^1(f_1^\rho, r) \quad = \quad & R(f_1^\rho) + (1/\lambda)[(1/n)\log E_{r(\gamma|\theta_0)f_1^\rho(\theta_0)}[((z_1(\gamma)f_1^\rho(\theta_0|\gamma))/r(\gamma|\theta_0))^n] - \\ & E_{f_1^\rho}[\log(f_1^\pi(\theta_0))] + \log(1/\delta) + (\lambda^2/M)] \qquad (6)\end{aligned}$$

where n is a parameter of the bound. We note however that if a Monte-Carlo estimator is used for the expectation in the second term, this gives a biased estimate of the CUBO bound [13], and hence only an approximate upper-bound on the expected risk.

In Eqs. 4 and 6, the cross-entropy terms $E_{f_1^\rho}[\log(f_1^\pi(\theta_0))]$ may be evaluated through Monte-Carlo approximation, that is, sampling from z_1, and evaluating $f_1^\pi(\theta_0|z_1)$ via the reparameterization trick, for samples θ_0 drawn from f_1^ρ. Otherwise, we can introduce a further ELBO bound on $\log(f_1^\pi(\theta_0))$, and evaluate this for samples from f_1^ρ. Both of these approaches preserve the upper-bound on the expected risk; we use the former in our experimentation. We note that all three approaches above can also be applied in the context of the 2-level PAC-Bayes bound (Eq. 3); again, we decompose the KL terms and use the approach of [14] to upper-bound the resulting entropy terms in our experimentation, as in Eq. 4. Finally, we note that although Eq. 4 introduces two variational distributions, these may be tied by setting $r_2 = r_1(\text{samp } r_{2a}(x,y))$, for $r_{2a} : (X,Y) \rightarrow \Theta_0'$, so that they share a consistent model of the mapping from parameters θ_0 to the latent space Z^S. The expansion of the 2-level bound introduces further variational distributions which may also be tied in this way, offering the possibility for efficient amortized inference (i.e. allowing inference models to jointly constrain each other). We leave experimental investigation of such tied variational bounds for future work.