PAC-Bayesian Hypernetworks

Pascal Germain

www.pascalgermain.info

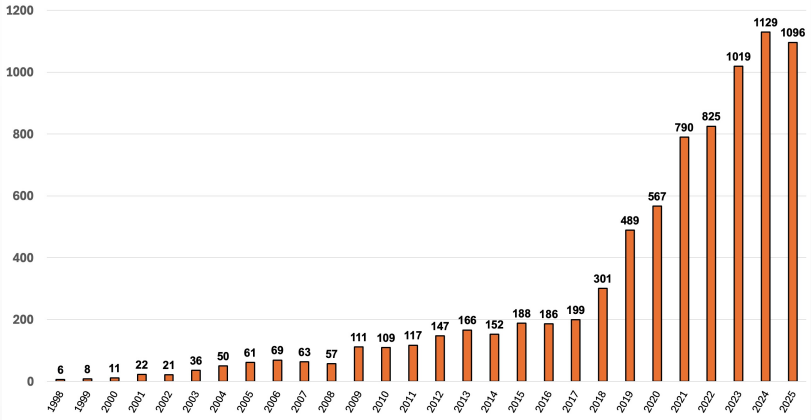
Département d'informatique et de génie logiciel

Université Laval

October 28, 2025

PAC-Bayesian Theory

Pioneered by
Shawe-Taylor and Williamson (1997), McAllester (1999), and Catoni (2003).



Number of search results per year for “PAC-Bayes(ian)” keywords on Google Scholar.

This talk: **The Mechanization of PAC-Bayes**

Computer scientists [...] must create abstractions of real-world problems that can be understood by computer users and, at the same time, that can be represented and manipulated inside a computer.

— *Foundations of Computer Science (Aho and Ullman 1992).* .
Chapter 1. **Computer Science: The Mechanization of Abstraction**

- Focus on PAC-Bayes for machine learning algorithm design;
- Propose to craft neural network architectures inspired from learning theory principles;
- A (perfectible) step towards guarantees for nowadays large/multimodal/foundation "models".

Plan

Definitions

A **learning example** $z := (x, y) \in \mathcal{X} \times \mathcal{Y} = \mathcal{Z}$ is a **description-label** pair.

Data generating distribution

Each example is an **observation from distribution** D on $\mathcal{Z}$.

Learning sample

$$S := \{ z_1, z_2, \ldots, z_n \} \sim D^n$$

Predictors (or hypothesis)

$$h : \mathcal{X} \to \mathcal{Y}, \quad h \in \mathcal{H}$$

Learning algorithm

$$A(S) \longrightarrow h$$

Loss function

$$\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$$

Empirical loss

$$\widehat{\mathcal{L}}_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, z_i)$$

Generalization loss

$$\mathcal{L}_D(h) = \mathop{\mathbf{E}}_{z \sim D} \ell(h, z)$$

A *Classical* PAC-Bayesian Theorem

PAC-Bayesian theorem (adapted from McAllester 1999; McAllester 2003)

For any distribution P on $\mathcal{H}$, for any $\delta \in (0,1]$, we have,

$$\Pr_{S\sim D^n}\left(\forall\, Q \text{ on } \mathcal{H} : \underbrace{\mathop{\mathbf{E}}_{h\sim Q}\mathcal{L}_D(h)}_{\substack{\text{generalization}\\ \text{loss}}} \leq \underbrace{\mathop{\mathbf{E}}_{h\sim Q}\widehat{\mathcal{L}}_S(h)}_{\text{empirical loss}} + \underbrace{\sqrt{\tfrac{1}{2n}\left[\mathrm{KL}(Q\|P) + \ln\tfrac{2\sqrt{n}}{\delta}\right]}}_{\text{complexity term}}\right) \geq 1-\delta\,,$$

where $\mathrm{KL}(Q\|P) = \mathop{\mathbf{E}}_{f\sim Q} \ln \frac{Q(f)}{P(f)}$ is the **Kullback-Leibler divergence**.

Valid *for all* posterior Q on $\mathcal{H}$

- Inspiration for conceiving **new learning algorithms**.

Tighter bounds for the [0, 1]-loss (Classical PAC-Bayes theorems)

$$\mathrm{kl}\Big(\mathop{\mathbf{E}}_{h\sim Q}\widehat{\mathcal{L}}_S(h),\ \mathop{\mathbf{E}}_{h\sim Q}\mathcal{L}_D(h)\Big) \ \leq\ \tfrac{1}{n}\left[\mathrm{KL}(Q\|P)+\ln\tfrac{2\sqrt{n}}{\delta}\right] \qquad \mathrm{kl}(q,p)\coloneqq q\ln\tfrac{q}{p}+(1-q)\ln\tfrac{1-q}{1-p}$$

From an algorithm design perspective, the "kl bound" promotes the minimization of

$$\mathrm{kl}^{-1}\left(\mathop{\mathbf{E}}_{h\sim Q}\widehat{\mathcal{L}}_S(h),\ \tfrac{1}{n}\left[\mathrm{KL}(Q\|P)+\ln\tfrac{2\sqrt{n}}{\delta}\right]\right)\coloneqq \operatorname*{argsup}_{0\leq p\leq 1}\left\{\mathrm{kl}\Big(\mathop{\mathbf{E}}_{h\sim Q}\widehat{\mathcal{L}}_S(h),p\Big)\ \leq\ \tfrac{1}{n}\left[\mathrm{KL}(Q\|P)+\ln\tfrac{2\sqrt{n}}{\delta}\right]\right\}$$

The function kl^{-1} is differentiable (see Reeb et al. 2018)

pyTorch implementation (Viallard et al. 2021):
https://github.com/paulviallard/ECML21-PB-CBound/blob/master/core/kl_inv.py

Lemma (see Letarte et al. 2019)

$$\mathrm{kl}^{-1}\left(\mathop{\mathbf{E}}_{h\sim Q}\widehat{\mathcal{L}}_S(h),\ \tfrac{1}{n}\left[\mathrm{KL}(Q\|P)+\ln\tfrac{2\sqrt{n}}{\delta}\right]\right)=\inf_{c>0}\left\{\frac{1}{1-e^{-c}}\left(c\cdot\mathop{\mathbf{E}}_{h\sim Q}\widehat{\mathcal{L}}_S(h)+\tfrac{1}{n}\left[\mathrm{KL}(Q\|P)+\ln\tfrac{2\sqrt{n}}{\delta}\right]\right)\right\}$$

Distribution over parameters

Given a model / predictor h_θ, where θ are parameters.

Consider P and Q as distributions over the set of parameters Θ.

$$\forall\, Q \text{ on } \Theta: \quad \mathrm{kl}\Big(\mathop{\mathbf{E}}_{\theta\sim Q}\widehat{\mathcal{L}}_S(h_\theta),\ \mathop{\mathbf{E}}_{\theta\sim Q}\mathcal{L}_D(h_\theta)\Big) \leq \frac{1}{n}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{n}}{\delta}\right].$$

Typical approach for (stochastics) neural networks

(Dziugaite and Roy 2017; Neyshabur, Bhojanapalli, and Srebro 2018; Nozawa, Germain, and Guedj 2020; Pérez-Ortiz et al. 2021, among many others.)

- $P = \mathcal{N}(\mathbf{W}_p, \sigma_p \mathbf{I})$ where $\mathbf{W}_p$ are the random/pre-learned weights initialization.
- $Q = \mathcal{N}(\mathbf{W}, \sigma \mathbf{I})$, where $\mathbf{W}$ are the learned/fine-tuned neural network weights.

Then, $\mathrm{KL}(Q\|P) = \frac{1}{2}\|\mathbf{W} - \mathbf{W}_p\|^2$.

Self-certified learning of neural networks (Pérez-Ortiz et al. 2021)

Pérez-Ortiz, Rivasplata, Shawe-Taylor and Szepesvári

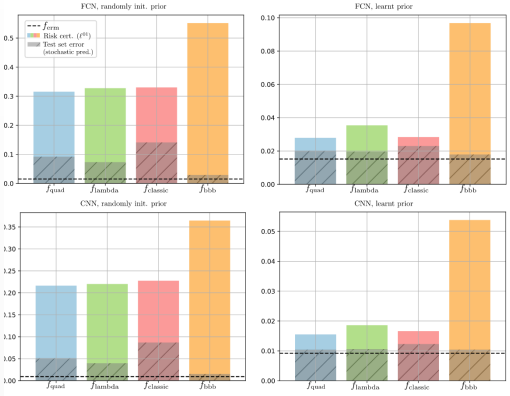


Figure 3: Tightness of the risk certificates for MNIST across different architectures, priors and training objectives. The bottom shaded areas correspond to the test set 0-1 error of the stochastic classifier. The coloured areas on top indicate the tightness of the risk certificate (smaller is better). The horizontal dashed line corresponds to the test set 0-1 error of f_{erm}, i.e. the deterministic classifier learnt

- Build on the pioneer work of Dziugaite and Roy (2017).
- Tight guarantees!

risk $\leq 1.55\%$ **on MNIST (CNN)** with probability $\geq 95\%$.

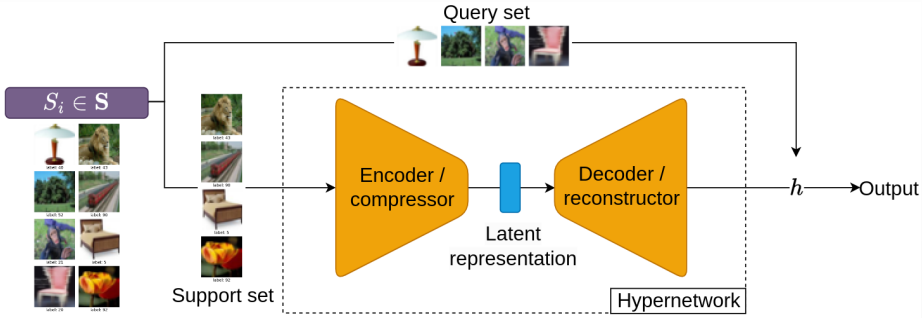
- Easy to train.

Source code (pyTorch):
https://github.com/mperezortiz/PBB

Plan

Overview of our meta-learning framework

Benjamin Leblanc et al. (2025). "Generalization Bounds via Meta-Learned Model Representations: PAC-Bayes and Sample Compression Hypernetworks". In: *ICML*. PMLR



Definitions

Meta-Learning dataset

$$\mathcal{S} \coloneqq \{S_1, S_2, \ldots, S_m\} \text{ such that } S_i \sim \left(D_i\right)^{n_i}.$$

Meta-Learning algorithm

$$\mathcal{A}(\mathcal{S}) \longrightarrow (\phi, \psi)$$

Encoder

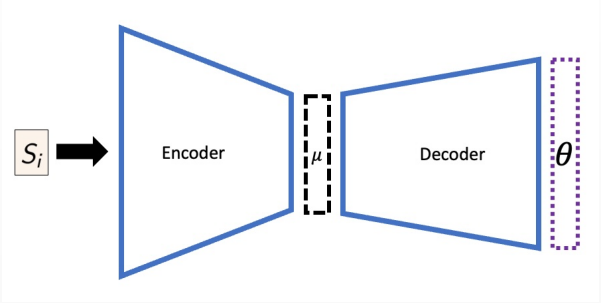
$$\mathcal{E}_\phi(S) \longrightarrow \boldsymbol{\mu}$$

Decoder

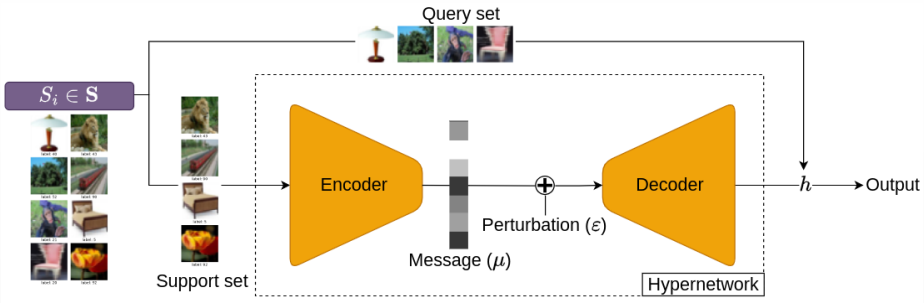
$$\mathcal{D}_\psi(\boldsymbol{\mu}) \longrightarrow \theta$$

(Downstream) Predictor

$$h_\theta(\mathbf{x}) \longrightarrow y$$



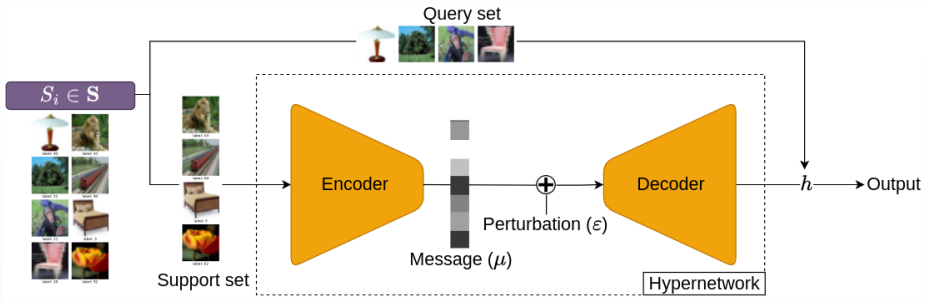
PAC-Bayes Hypernetworks



Training objective. Each dataset $S_i \in \boldsymbol{\mathcal{S}}$ is split into a *support set* $\hat{S}_i$ and a *query set* $\hat{T}_i = S_i \backslash \hat{S}_i$.

$$\min_{\psi,\phi} \left\{ \frac{1}{m} \sum_{i=1}^{m} \mathbf{E}\, \widehat{\mathcal{L}}_{\hat{T}_i}(h_{\theta_i}) \;\middle|\; \theta_i = \mathcal{D}_\psi(\boldsymbol{\mu}_i + \boldsymbol{\epsilon});\ \boldsymbol{\mu}_i = \mathcal{E}_\phi(\hat{S}_i) \right\}, \qquad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

PAC-Bayes Hypernetworks

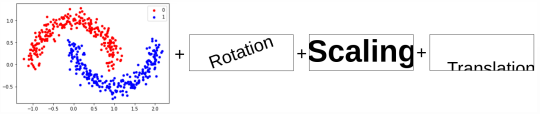


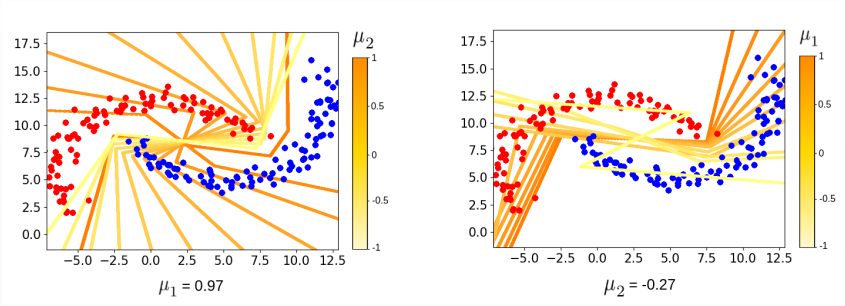
Bound computation. Given a new task sample $S' \sim (\mathcal{D}')^{n'}$, let $\boldsymbol{\mu} = \mathcal{E}_\phi(S')$ and $\theta' = \mathcal{D}_\psi(\boldsymbol{\mu} + \boldsymbol{\epsilon})$.

$$\mathbf{E}\,\mathcal{L}_{D'}(h_{\theta'}) \leq \operatorname*{argsup}_{0 \leq p \leq 1} \left\{ \mathrm{kl}\Big(\mathbf{E}\,\widehat{\mathcal{L}}_{S'}(h_{\theta'}), p \Big) \leq \frac{\frac{1}{2}\|\boldsymbol{\mu}\|^2 + \ln \frac{2\sqrt{n'}}{\delta}}{n'} \right\},$$

using a prior $P_{\mathbf{0}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a posterior $Q = \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$ over the latent representation space.

Toy Experiments





$$\theta = \mathcal{D}_\psi([\mu_1, \mu_2]^T)$$

MNIST pixel swap

- Multiclass classification
- Meta-train set: 10 tasks of 60 000 examples
- Meta-test set: 20 tasks of 2000 examples

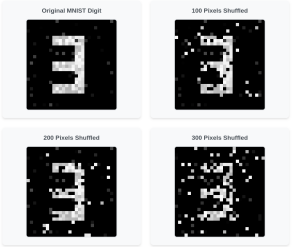
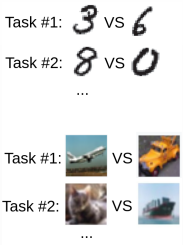


Image: Benjamin Leblanc & Claude

Algorithm	100 Pixels swap		200 Pixels swap		300 Pixels swap	
	Bound	Test error	Bound	Test error	Bound	Test error
Pentina and Lampert (2014)	0.190	0.019	0.240	0.026	0.334	0.038
Amit and Meir (2018)	0.138	0.016	0.161	0.020	0.329	0.040
Guan and Lu (2022)	0.093	**0.015**	0.128	**0.019**	0.210	**0.024**
Zakerinia, Behjati, and Lampert (2024)	**0.053**	0.019	**0.108**	0.026	**0.149**	0.035
Our PAC-Bayes Hypernetwork	0.068	0.027	0.112	0.076	0.219	0.186
Opaque encoder		0.037		0.087		0.159

MINIST and CIFAR100 Binary Pairs

- Binary classification
- Meta-train set: 56 tasks of 2000 examples for MNIST;
  100 tasks of 1200 examples for CIFAR
- Meta-test set: 34 tasks of 2000 examples for MNIST;
  50 tasks of 200 examples for CIFAR



Algorithm	MNIST		CIFAR100	
	Bound	Test error	Bound	Test error
Pentina and Lampert (2014)	0.767 ± 0.001	0.369 ± 0.223	0.801 ± 0.001	0.490 ± 0.070
Amit and Meir (2018)	1372 ± 23.36	0.351 ± 0.212	950.9 ± 343.1	0.284 ± 0.120
Guan and Lu (2022)	0.754 ± 0.003	0.366 ± 0.221	**0.802** ± 0.001	0.489 ± 0.073
Zakerinia, Behjati, and Lampert (2024)	0.684 ± 0.021	0.351 ± 0.212	0.953 ± 0.315	**0.281** ± 0.125
Our PAC-Bayes Hypernetwork	**0.597** ± 0.107	**0.150** ± 0.114	0.974 ± 0.022	0.295 ± 0.103
Opaque encoder		0.497 ± 0.134		0.506 ± 0.101

Plan

History

Inception: Littlestone and Warmuth (1986): "Relating Data Compression and Learnability"

The Set Covering Machine

Marchand and Shawe-Taylor (2002): "The Set Covering Machine"
Marchand and Sokolova (2005): "Learning with Decision Lists of Data-Dependent Features"
Laviolette, Marchand, and Shah (2005): "Margin-Sparsity Trade-Off for the Set Covering Machine"
Hussain et al. (2007): "Revised Loss Bounds for the Set Covering Machine and Sample-Compression Loss Bounds for Imbalanced Data"
Drouin et al. (2019): "Interpretable genotype-to-phenotype classifiers with performance guarantees"

Pick-To-Learn

Campi and Garatti (2023): "Compression, Generalization and Learning"
Paccagnan, Campi, and Garatti (2023): "The Pick-to-Learn Algorithm: Empowering Compression for Tight Generalization Bounds and Improved Post-training Performance"
Marks and Paccagnan (2025): "Pick-to-Learn and Self-Certified Gaussian Process Approximations"
Bazinet, Zantedeschi, and Germain (2025): "Sample Compression Unleashed: New Generalization Bounds for Real Valued Losses"

Definitions

A **compressed predictor** $h_{\mathbf{i}}^{\mu}$ is a data-dependant predictor encoded by two quantities

- A **compression set** $S_{\mathbf{i}}$ is a subset of the training set S:
  - $\mathbf{i} \in \mathcal{I}_n \coloneqq \mathcal{P}(\{1, 2, \ldots, n\})$
- A **message** $\mu \in \mathcal{M}_{\mathbf{i}}$ contains additional information to describe the predictor $h_{\mathbf{i}}^{\mu}$.
  - The message μ is chosen among a (discrete) set $\mathcal{M}_{\mathbf{i}}$ of predefined messages given $S_{\mathbf{i}}$.
  - For simplicity, we sometime use a message set $\mathcal{M}$ that does not rely on $S_{\mathbf{i}}$.

Given $S_{\mathbf{i}} \in \mathcal{Z}^{|\mathbf{i}|}$ et $\mu \in \mathcal{M}_{\mathbf{i}}$, a **reconstruction function** $\mathcal{R}$ outputs a predictor :

$$h_{\mathbf{i}}^{\mu} \;=\; \mathcal{R}(S_{\mathbf{i}}, \mu)\,.$$

A Classical Sample Compressed Classifiers

SVM: Support Vector Machine (hard margin)

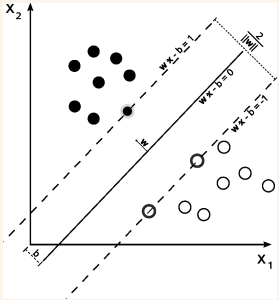


Image: Wikipedia

The SVM's learning algorithm acts as its own reconstruction function

$$\mathrm{SVM}(S) = h_{\mathbf{i}}^{\mu} = \mathrm{SVM}(S_{\mathbf{i}})$$

with $S_{\mathbf{i}} = \{\text{support vectors}\}$
and $\mu = \emptyset$

Generalization Bounds for sample compressed binary classifiers

Given $h^{\mu}_{\mathbf{i}} : \mathcal{X} \to \{-1, +1\}$, and $\mathcal{L}^{01}_{D}(h^{\mu}_{\mathbf{i}}) = \mathop{\mathbf{E}}_{(x,y)\sim D} I(h^{\mu}_{\mathbf{i}}(x) \neq y)$ the zero-one loss.

Theorem (Marchand and Sokolova 2005; Laviolette, Marchand, and Shah 2005)

Let $\mathcal{R}$ be a reconstruction function, $P_{\mathcal{M}_{\mathbf{i}}}$ a distribution over messages, and $\delta \in (0, 1]$. With high probability ($\geq 1 - \delta$) over $S \sim D^n$, we have

$\forall \mathbf{i} \in \mathcal{I}_n, \mu \in \mathcal{M}_{\mathbf{i}}$:

$$\mathcal{L}^{01}_{D}(h^{\mu}_{\mathbf{i}}) \leq 1 - \exp\left(\frac{-1}{n - |\mathbf{i}| - k_{S_{\mathbf{i}^c}}}\left[\ln\binom{n - |\mathbf{i}|}{k_{S_{\mathbf{i}^c}}} + \ln\binom{n}{|\mathbf{i}|} + \ln\left(\frac{1}{P_{\mathcal{M}_{\mathbf{i}}}(\mu) \cdot \xi(|\mathbf{i}|) \cdot \delta}\right)\right]\right)$$

where $k_{S_{\mathbf{i}^c}} := |\mathbf{i}^c| \widehat{\mathcal{L}}^{01}_{S_{\mathbf{i}^c}}(h^{\mu}_{\mathbf{i}})$ is the error count on $S_{\mathbf{i}^c} := S \setminus S_{\mathbf{i}}$ and $\xi(a) := \frac{6}{\pi^2}(a+1)^{-2}$.

New bound for real-valued losses

Assumption

The loss $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ is σ-sub-Gaussian, *i.e.*, for all $\mathbf{i} \in \mathcal{I}_n, \mu \in \mathcal{M}_{\mathbf{i}}$:

$$\mathop{\mathbf{E}}_{z\sim D} e^{\lambda(\ell(h_{\mathbf{i}}^{\mu},z)-\mathcal{L}_D(h_{\mathbf{i}}^{\mu}))} \leq e^{\frac{\lambda^2\sigma^2}{2}}, \quad \forall \lambda \in \mathbb{R}.$$

Theorem

Let $\mathcal{R}$ be a reconstruction function, $P_{\mathcal{M}_{\mathbf{i}}}$ a distribution over messages, and $\delta \in (0, 1]$. With high probability ($\geq 1 - \delta$) over $S \sim D^n$, we have

$\forall \mathbf{i} \in \mathcal{I}_n, \mu \in \mathcal{M}_{\mathbf{i}}$:

$$\mathcal{L}_D(h_{\mathbf{i}}^{\mu}) \leq \widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}^{\mu}) + \frac{1}{\sqrt{n-|\mathbf{i}|}}\left[\frac{\sigma^2}{2} + \ln\binom{n}{|\mathbf{i}|} + \ln\left(\frac{1}{P_{\mathcal{M}_{\mathbf{i}}}(\mu)\cdot\xi(|\mathbf{i}|)\cdot\delta}\right)\right].$$

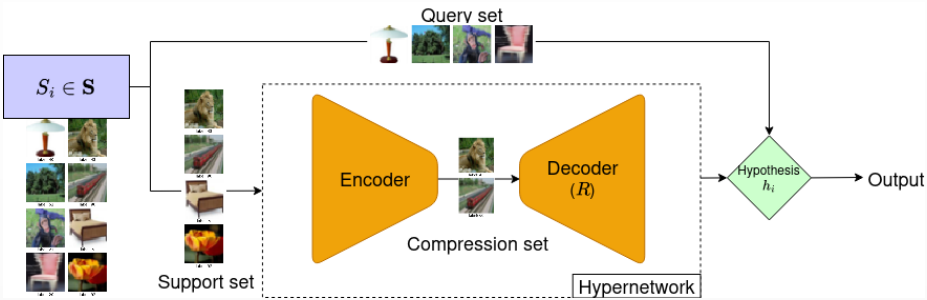
Proof idea. Each sample compressed classifier $h_{\mathbf{i}}^{\mu}$ is defined independently of $S_{\mathbf{i}^c} \sim D^{n-|\mathbf{i}|}$.

For all $\mathbf{i} \in \mathcal{I}_n, \mu \in \mathcal{M}_{\mathbf{i}}$ and $\delta_{\mathbf{i},\mu} \in (0,1]$, with probability at least $1-\delta_{\mathbf{i},\mu}$:

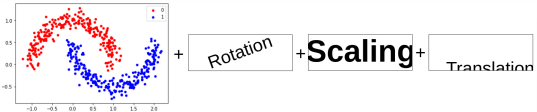
$$
\begin{aligned}
\mathcal{L}_D(h_{\mathbf{i}}^{\mu}) - \widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}^{\mu}) &\leq \frac{1}{t}\left[\ln\left(\mathop{\mathbf{E}}_{S_{\mathbf{i}^c}\sim D^{n-|\mathbf{i}|}} e^{t\left(\mathcal{L}_D(h_{\mathbf{i}}^{\mu}) - \widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}^{\mu})\right)}\right) + \ln\frac{1}{\delta_{\mathbf{i},\mu}}\right] \quad \langle \text{ Chernoff } (t>0) \rangle \\
&= \frac{1}{t}\left[\ln\left(\prod_{i=1}^{n-|\mathbf{i}|}\mathop{\mathbf{E}}_{z\sim D} e^{\frac{t}{n-|\mathbf{i}|}\left(\ell(h_{\mathbf{i}}^{\mu},z) - \widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}^{\mu})\right)}\right) + \ln\frac{1}{\delta_{\mathbf{i},\mu}}\right] \\
&\leq \frac{1}{t}\left[\ln\left(\prod_{i=1}^{n-|\mathbf{i}|} e^{\frac{t^2\sigma^2}{2(n-|\mathbf{i}|)^2}}\right) + \ln\frac{1}{\delta_{\mathbf{i},\mu}}\right] \quad \langle \text{ sub-Gaussian loss, with } \lambda := \frac{t}{n-|\mathbf{i}|} \rangle \\
&= \frac{1}{t}\left[\frac{t^2\sigma^2}{2(n-|\mathbf{i}|)} + \ln\frac{1}{\delta_{\mathbf{i},\mu}}\right].
\end{aligned}
$$

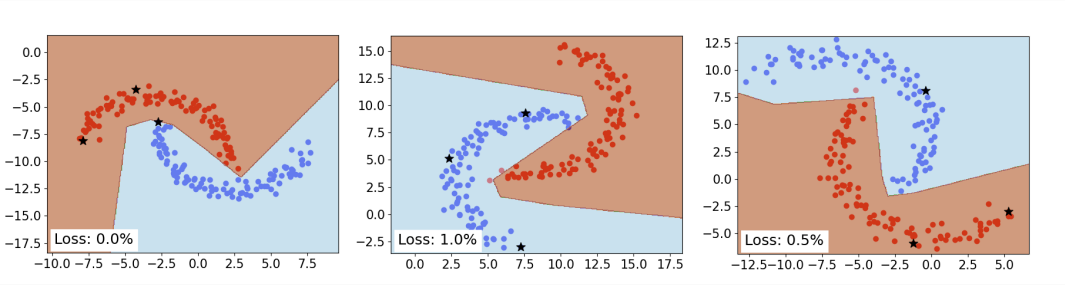
The final result is obtained by choosing $t := \sqrt{n-|\mathbf{i}|}$ and by an union bound over the (discrete) set of all possible sample compress classifiers, with $\delta_{\mathbf{i},\mu} = \frac{\xi(|\mathbf{i}|)}{\binom{n}{|\mathbf{i}|}} \cdot P_{\mathcal{M}_{\mathbf{i}}}(\mu) \cdot \delta$, since $\sum_{i=1}^{\infty}\xi(i) = 1$.

Sample Compress Hypernetworks



Toy Experiments





Plan

New bound for continuous messages

PAC-Bayes to the rescue!

We now consider a posterior distribution $Q_{\mathcal{M}}$ over the (possibly continuous) message set $\mathcal{M}$.

Theorem

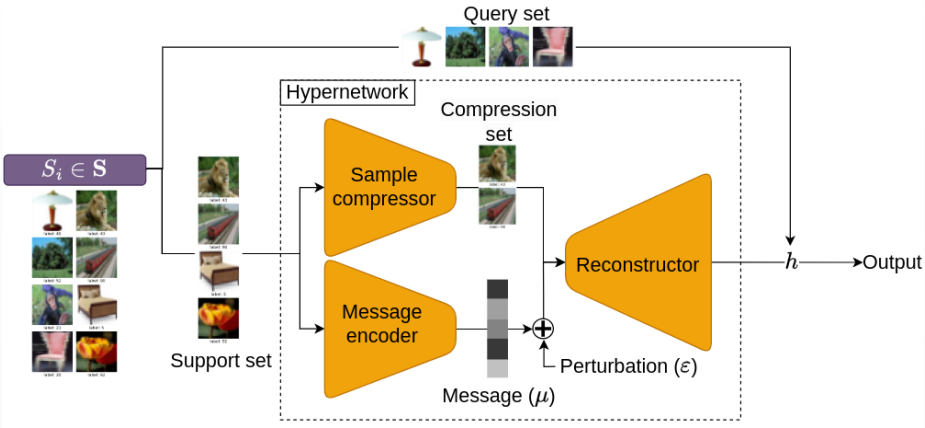
Let $\mathcal{R}$ be a reconstruction function, $P_{\mathcal{M}}$ a distribution over messages, and $\delta \in (0, 1]$. With high probability ($\geq 1 - \delta$) over $S \sim D^n$, we have

$\forall \mathbf{i} \in \mathcal{I}_n, Q_{\mathcal{M}}$ over $\mathcal{M}$:

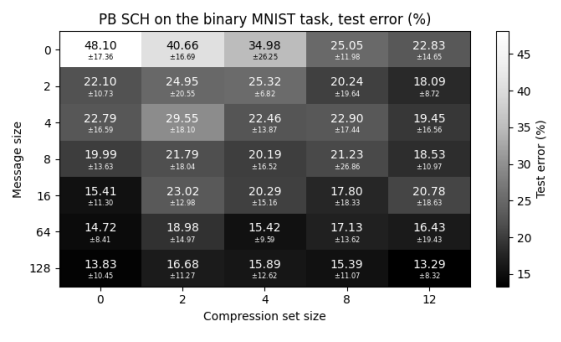
$$\mathop{\mathbf{E}}_{\mu \sim Q_{\mathcal{M}}} \mathcal{L}_D(h_{\mathbf{i}}^{\mu}) \leq \mathop{\mathbf{E}}_{\mu \sim Q_{\mathcal{M}}} \widehat{\mathcal{L}}_{S_{\overline{\mathbf{i}}^c}}(h_{\mathbf{i}}^{\mu}) + \frac{1}{\sqrt{n - |\mathbf{i}|}} \left[\mathrm{KL}(Q_{\mathcal{M}} \| P_{\mathcal{M}}) + \frac{\sigma^2}{2} + \ln \binom{n}{|\mathbf{i}|} + \ln \left(\frac{1}{\xi(|\mathbf{i}|) \cdot \delta} \right) \right].$$

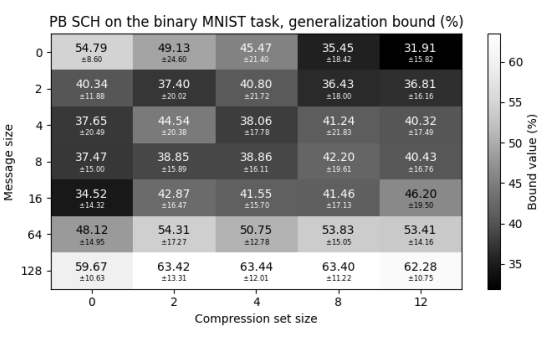
Proof idea. For a fixed $\mathbf{i} \in \mathcal{I}_n$, get a typical PAC-Bayes bound with prior/posterior distribution over messages $\mathcal{M}$. Then, use a **union bound** to get a bound uniformly valid over the compression sets $\mathcal{I}_n$.

PAC-Bayes Sample Compress Hypernetworks



MNIST Binary Pairs





Plan

Perspectives

- Dynamically select the message/compression set size during learning;
- Fine-tune large language models.

Collaborators

Benjamin Leblanc

Mathieu Bazinet

Nathaniel D'Amours

Alexandre Drouin

- Aho, Alfred V. and Jeffrey D. Ullman (1992). *Foundations of Computer Science, C Edition*. Computer Science Press / W. H. Freeman.
- Amit, Ron and Ron Meir (2018). "Meta-Learning by Adjusting Priors Based on Extended PAC-Bayes Theory". In: *ICML*.
- Bazinet, Mathieu, Valentina Zantedeschi, and Pascal Germain (2025). "Sample Compression Unleashed: New Generalization Bounds for Real Valued Losses". In: *AISTATS*. Vol. 258. Proceedings of Machine Learning Research. PMLR, pp. 3286–3294.
- Campi, Marco C. and Simone Garatti (2023). "Compression, Generalization and Learning". In: *JMLR* abs/2301.12767.
- Catoni, Olivier (2003). "A PAC-Bayesian approach to adaptive classification". In: *preprint* 840.
- Drouin, Alexandre et al. (2019). "Interpretable genotype-to-phenotype classifiers with performance guarantees". In: *Scientific reports* 9.1, pp. 1–13.
- Dziugaite, Gintare Karolina and Daniel M. Roy (2017). "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data". In: *UAI*. AUAI Press.
- Guan, Jiechao and Zhiwu Lu (2022). "Fast-Rate PAC-Bayesian Generalization Bounds for Meta-Learning". In: *ICML*.

- Hussain, Zakria et al. (2007). "Revised Loss Bounds for the Set Covering Machine and Sample-Compression Loss Bounds for Imbalanced Data". In: *J. Mach. Learn. Res.* 8, pp. 2533–2549.
- Laviolette, François, Mario Marchand, and Mohak Shah (2005). "Margin-Sparsity Trade-Off for the Set Covering Machine". In: *ECML*. Vol. 3720. Lecture Notes in Computer Science. Springer, pp. 206–217.
- Leblanc, Benjamin et al. (2025). "Generalization Bounds via Meta-Learned Model Representations: PAC-Bayes and Sample Compression Hypernetworks". In: *ICML*. PMLR.
- Letarte, Gaël et al. (2019). "Dichotomize and Generalize: PAC-Bayesian Binary Activated Deep Neural Networks". In: *NeurIPS*, pp. 6869–6879.
- Littlestone, Nick and Manfred K. Warmuth (1986). "Relating Data Compression and Learnability". In: *Technical Report*.
- Marchand, Mario and John Shawe-Taylor (2002). "The Set Covering Machine". In: *JMLR* 3.
- Marchand, Mario and Marina Sokolova (2005). "Learning with Decision Lists of Data-Dependent Features". In: *JMLR* 6.

- Marks, Daniel and Dario Paccagnan (2025). "Pick-to-Learn and Self-Certified Gaussian Process Approximations". In: *AISTATS*. Vol. 258. Proceedings of Machine Learning Research. PMLR, pp. 2656–2664.
- McAllester, David (1999). "Some PAC-Bayesian Theorems". In: *Machine Learning* 37.3.
- — (2003). "PAC-Bayesian Stochastic Model selection". In: *Machine Learning* 51.1.
- Neyshabur, Behnam, Srinadh Bhojanapalli, and Nathan Srebro (2018). "A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks". In: *ICLR (Poster)*. OpenReview.net.
- Nozawa, Kento, Pascal Germain, and Benjamin Guedj (2020). "PAC-Bayesian Contrastive Unsupervised Representation Learning". In: *UAI*. Vol. 124. Proceedings of Machine Learning Research. AUAI Press, pp. 21–30.
- Paccagnan, Dario, Marco C. Campi, and Simone Garatti (2023). "The Pick-to-Learn Algorithm: Empowering Compression for Tight Generalization Bounds and Improved Post-training Performance". In: *NeurIPS*.
- Pentina, Anastasia and Christoph H. Lampert (2014). "A PAC-Bayesian bound for Lifelong Learning". In: *ICML*.

- Pérez-Ortiz, María et al. (2021). "Tighter Risk Certificates for Neural Networks". In: *J. Mach. Learn. Res.* 22, 227:1–227:40.
- Reeb, David et al. (2018). "Learning Gaussian Processes by Minimizing PAC-Bayesian Generalization Bounds". In: *NeurIPS*, pp. 3341–3351.
- Shawe-Taylor, John and Robert C. Williamson (1997). "A PAC Analysis of a Bayesian Estimator". In: *COLT*.
- Viallard, Paul et al. (2021). "Self-bounding Majority Vote Learning Algorithms by the Direct Minimization of a Tight PAC-Bayesian C-Bound". In: *ECML/PKDD (2)*. Vol. 12976. Lecture Notes in Computer Science. Springer, pp. 167–183.
- Zakerinia, Hossein, Amin Behjati, and Christoph H. Lampert (2024). "More flexible PAC-Bayesian meta-learning by learning learning algorithms". In: *ICML*.