Hot training errors predict cold test errors in the interpolation regime

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the generalization of iterative learning algorithms, which follow the negative gradient of the empirical error and inject noise at every step. Identifying the quantity of injected randomness with temperature, we demonstrate theoretically and experimentally that the training errors of hotter limiting distributions are predictive of the test errors of colder limiting distributions.

1 The annealing principle

The main contributions of this work are theoretical arguments and experimental evidence to support the following statement:

> *For algorithms akin to Langevin dynamics, smaller training errors of hot limiting distributions indicate smaller test errors of cold limiting distributions.*

The algorithms we consider are stochastic gradient descent (SGD, Robbins and Monro [1951]), stochastic gradient Langevin dynamics (SGLD, Gelfand and Mitter [1991]), the unadjusted Langevin algorithm (ULA, Parisi [1981]) and their idealization, continuous Langevin dynamics (CLD, Langevin [1908]). Each of these, if evolving under constant conditions, is a time-homogeneous Markov process, which under mild assumptions converges to a stationary distribution. This distribution depends on the training data, the type of algorithm in question, and is parametrized by the amount of randomness injected at each step. For CLD and ULA, this randomness is explicit as isotropic Gaussian noise; for SGD it takes the form of random mini-batches, while SGLD uses both forms of noise (cf. Appendix F).

There is thus for each of these methods a one-parameter family $\{\nu(r) : r \geq 0\}$ of distributions, with an *inverse-randomness parameter* r. We speak of hot (more randomness injected) and cold (less randomness injected) limiting distributions of the same algorithm. As limiting distributions of CLD, there is also a corresponding family of hot or cold, but not directly observable Gibbs distributions, parametrized by the inverse temperature β. The underlying theory projects the ν-families to the Gibbs family, for which it exploits an explicit expression of the relative entropy to the prior (as in Maurer et al. [2025] *inside* the Gibbs family), and then uses the PAC-Bayesian theorem.

In Figure 1, we demonstrate our principle on 50 binary classification tasks of varying difficulty generated from the CIFAR-100 dataset, using a fully connected ReLU network. For each task and algorithm we plot the test error (vertical axis) of the cold distribution against the training errors (horizontal axis) of the cold distribution and the average training error of corresponding hot distributions. The cold training errors are not predictive, but the hot training errors are substantially more predictive for ULA and SGLD and to a lesser extent for SGD, which is furthest from the ideal Langevin dynamics. The principle is particularly interesting in the context of overparametrized neural networks, where *hotter distributions* are often *more efficiently generated*, and the cold distributions have near-zero training error, which is not predictive of the test error, as many experiments with randomly labelled data have shown (Zhang et al. [2021], see also Figure 2).

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

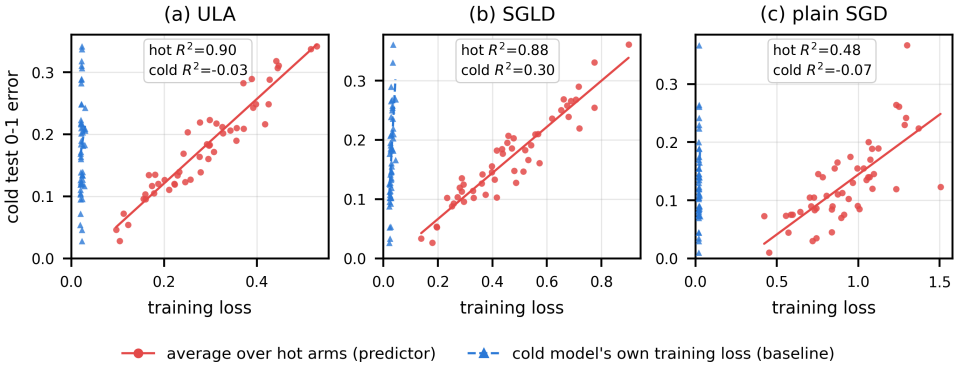


Figure 1: Cold test error against the average training error of hot (circles, solid fit) and cold (triangles, dashed fit) distributions, over 50 binary CIFAR-100 tasks: (a) ULA, (b) SGLD, (c) SGD.

2 Theory

In this section we explain the underlying theory. We first introduce some necessary notation. Let $\mathcal{X}$ be a space of *data* with distribution μ and $\mathbf{x} \sim \mu^n$ the training sample. $\mathcal{H}$ is a *hypothesis* space, $\mathcal{P}(\mathcal{H})$ is the set of probability measures on $\mathcal{H}$, $\ell : \mathcal{H} \times \mathcal{X} \to [0, \infty)$ is a loss function and $L(h) = \mathbb{E}_{x\sim\mu}[\ell(h, x)]$ and $\hat{L}(h, \mathbf{x}) = \frac{1}{n}\sum_i \ell(h, x_i)$ denote the population and empirical errors. For $\rho, \nu \in \mathcal{P}(\mathcal{H})$ the relative entropy is denoted $\mathrm{KL}(\rho, \nu)$. We fix a data-independent prior $\pi \in \mathcal{P}(\mathcal{H})$ with $\mathbb{E}_\pi[\hat{L}] < \infty$, $\mathbb{E}_\pi[\hat{L}^2] < \infty$, and $\hat{L}$ non-constant π-almost surely (regularity conditions detailed in Appendix B.1). We write $\mathbb{E}_\nu[\hat{L}]$, $\mathbb{V}_\nu[\hat{L}]$ for the mean and variance of the empirical error under ν, omitting the dependence on $\mathbf{x}$, whenever there is no ambiguity.

The PAC-Bayesian Theorem. We state the theorem for losses with values in $[0, 1]$. It follows from a general version given by Rivasplata et al. [2020]. Versions for sub-Gaussian losses and all proofs are given in Appendix C.

Theorem 1. *Let $\nu : \mathcal{X}^n \to \mathcal{P}(\mathcal{H})$ be a stochastic algorithm such that $\nu(\mathbf{x})$ is absolutely continuous w.r.t. π for all $\mathbf{x} \in \mathcal{X}^n$. Then with probability at least $1 - \delta$ as $\mathbf{x} \sim \mu^n$*

$$\kappa\left(\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[\hat{L}\left(h, \mathbf{x}\right)\right], \mathbb{E}_{h\sim\nu(\mathbf{x})}\left[L\left(h\right)\right]\right) \leq \frac{\mathrm{KL}\left(\nu(\mathbf{x}), \pi\right) + \ln\left(2\sqrt{n}/\delta\right)}{n},$$

Here $\kappa(p, q) = p \ln \frac{p}{q} + (1 - p) \ln \frac{1-p}{1-q}$ (the relative entropy of two Bernoulli variables) is used as a measure of the generalization gap. In applications of the theorem we can restrict to distributions $\nu(\mathbf{x}) \in \mathcal{P}_\pi(\mathcal{H}) := \{\nu \in \mathcal{P}(\mathcal{H}) : \mathrm{KL}(\nu, \pi) < \infty, \mathbb{E}_\nu[\hat{L}] \leq \mathbb{E}_\pi[\hat{L}]\}$, for otherwise either the bound is infinite or we can simply use π as a better posterior.

Whichever variant of the result we use, the dominant term in the bound is $\mathrm{KL}(\nu(\mathbf{x}), \pi)$, which becomes the central object controlling generalization. When bounding this quantity it is conditional on the training data $\mathbf{x}$, explicit mention of which can be omitted.

Gibbs distributions. The Gibbs distribution at inverse temperature $\beta \geq 0$ is the member $G(\beta) \in \mathcal{P}(\mathcal{H})$ defined by

$$G(\beta)(A) = \frac{1}{Z(\beta)} \int_A e^{-\beta\hat{L}(h)} d\pi(h) \text{ for } A \in \Omega.$$

The normalizing factor $Z(\beta) := \int_{\mathcal{H}} e^{-\beta\hat{L}(h)} d\pi(h)$ is called the *partition function*. The function $G : \beta \in [0, \infty) \mapsto G(\beta) \in \mathcal{P}$ is called the Gibbs family. We also write φ for the function $\varphi : \beta \in [0, \infty) \mapsto \mathbb{E}_{G(\beta)}[\hat{L}]$. For Gibbs distributions the PAC-Bayesian complexity term, $\mathrm{KL}(G(\beta), \pi)$, is proportional to an area under the graph of φ [Maurer et al., 2025]. Here we extend this principle *beyond* the Gibbs family, to any one-parameter family corresponding to the stationary laws of stochastic iterative learning algorithms, with an explicit residual term quantifying the distance to the Gibbs family.

Projecting to the Gibbs family. Now let $\nu : [0, \infty) \to \mathcal{P}_\pi(\mathcal{H})$ be a one-parameter family of probability measures, representing the limiting distributions of some iterative algorithm at different

levels of randomness. Think of $\nu(r)$ as distributions of decreasing stochasticity, for example with decreasing injected Gaussian noise, as for ULA, or with increasing mini-batch sizes or decreasing learning rates as for SGD, or hybrid versions like SGLD.

For $s \in [0, \infty)$ we write $\psi(s) = \mathbb{E}_{\nu(s)}[\hat{L}]$ for the training error of $\nu(s)$ and let $\Delta(s)$ be the distance (in the sense of relative entropy) of $\nu(s)$ to the Gibbs family, i.e. $\Delta(s) = \min_{\beta \geq 0} \mathrm{KL}(\nu(s), G(\beta))$. Since decreasing noise should improve the empirical error, we will assume the function ψ to be strictly decreasing. We also assume $\pi = \nu(0)$, which is the maximally random distribution in common with the Gibbs family.

Theorem 2. *There exists a bijective function $\alpha : [0, \infty) \to [0, \infty)$, such that*

(i) $\Delta(r) = \mathrm{KL}(\nu(r), G(\alpha(r)))$

(ii) $\psi(r) = \varphi(\alpha(r))$

(iii) $\mathrm{KL}(\nu(r), \pi) = \Delta(r) + \mathrm{KL}(G(\alpha(r)), \pi) = \Delta(r) + \int_0^{\alpha(r)} (\varphi(\gamma) - \psi(r))\, d\gamma$

(iv) There is a probability measure ρ on the interval $[0, r]$ such that

$$\mathrm{KL}(\nu(r), \pi) = \Delta(r) + \alpha(r)\left(\mathbb{E}_{s\sim\rho}[\psi(s)] - \psi(r)\right). \tag{1}$$

(v) If $\hat{L}$ has values in $[0, 1]$, if ψ is continuously differentiable, and $\mathbb{V}_{\nu(s)}[\hat{L}] > \sqrt{\Delta(s)/2}$ for all $s \leq r$, then

$$\mathrm{KL}(\nu(r), \pi) \leq \Delta(r) + \int_0^r (\psi(s) - \psi(r)) \frac{(-\psi'(s))}{\mathbb{V}_{\nu(s)}\left[\hat{L}\right] - \sqrt{\Delta(s)/2}} ds.$$

Proof and illustrating figure are in Appendix D. By the PAC-Bayesian theorem, the quantity to bound is the KL-divergence from $\nu(r)$ to the prior $\pi = \nu(0)$. By (i) and (iii) this decomposes as

$$\mathrm{KL}(\nu(r), \pi) = \mathrm{KL}(\nu(r), G(\alpha(r))) + \mathrm{KL}(G(\alpha(r)), \pi).$$

If the KL-divergence was the square of a Euclidean distance (which it isn't), then this would be the Pythagorean theorem. The first term on the right hand side is the residual error $\Delta(r)$, quantifying the distance from the Gibbs family. The second term has an integral representation as in [Maurer et al., 2025]. By (ii) the empirical errors of $G(\gamma)$ and $\nu\left(\alpha^{-1}(\gamma)\right)$ are the same, so the bijection α can pull the scaled Lebesgue measure back to the interval $[0, r]$, as expressed in part (iv). Part (v) finally gives an explicit bound, with the residual Δ as the only unobservable quantity.

The Theorem is the basis of a simple strategy of relative prediction. Suppose that for some fixed algorithm we have generated the distributions $\{\nu_k(r) : k \in [K]\}$ for different learning tasks, which we assume to be at similar distances $\Delta_k(r)$ to the respective Gibbs families. These will be our cold distributions, with small empirical errors $\psi_k(r)$, which is however not predictive of the respective test errors. By part (iv) of Theorem 2 the PAC-Bayesian upper bounds on the respective generalization errors are increasing in $\mathbb{E}_{s\sim\rho}[\psi_k(s)]$. If we take the ρ_k as uniformly distributed (a maximal entropy assumption), we can draw $s_1, ..., s_J \sim \mathrm{Unif}[0, r]$ and estimate each expectation by averaging the $\psi_k(s_j)$ over j. The $\nu_k(s_j)$ are the "hot" distributions and their training errors are averaged to predict the respective test errors of the "cold" distributions $\nu_k(r)$, as in Figure 1. Of course, since the $\Delta_k(r)$ are unknown, these predictions are only relative among the tasks. Figure 9 as well as Table 1 are evidence that this crude scheme works already for a single value s_1.

3 Experiments

We evaluate our annealing principle in two settings. First, we compare correct and random labels on binary MNIST while varying the amount of algorithmic randomness in ULA, SGLD, and plain SGD. Second, we test whether hot training errors predict the generalization of a cold model across a collection of binary CIFAR-100 tasks.

Interpolation Puzzle: ULA, SGLD, and plain SGD. Figure 2 compares inverse-randomness paths on the same task. Panel (a) varies "β" for ULA, while panel (c) does the same for SGLD with batch size $B = 16$. Panel (b) shows plain SGD along a coupled batch-size/step-size ladder, and panel (d) fixes "β"$= 16000$ and varies the SGLD batch size. For ULA and SGLD, "β" is the algorithmic

target (cold)	predictor	R^2	Spearman	MAE	×MAE	grad.	wall
ULA β=$8n$	own training error (baseline)	-0.03	0.30	0.063	1.0	1.0	1.0
	ULA β=$n/4$	0.89	0.94	0.020	3.2	1.9	1.0
	SGLD β=$n/2$, B=16	0.90	0.94	0.020	3.2	135.4	4.8
GD (full batch)	own training error (baseline)	-0.11	0.60	0.053	1.0	1.0	1.0
	SGLD β=$n/2$, B=16	0.83	0.91	0.021	2.5	156.6	7.1
	SGD B=3	0.41	0.74	0.036	1.5	590.5	4.5

Table 1: Hot predictors against the cold models they predict, 50 binary CIFAR-100 tasks. R^2 is the leave-one-out coefficient of determination of the fitted affine map and Spearman the rank correlation, both across tasks; MAE is the mean absolute leave-one-out error in predicting cold test 0–1 error, and ×MAE how many times smaller it is than the baseline's. The last two columns are cost ratios, target to predictor: gradient evaluations and measured wall-clock. Appendix H.2 gives every predictor against every target, and Appendix G.6 defines the scores.

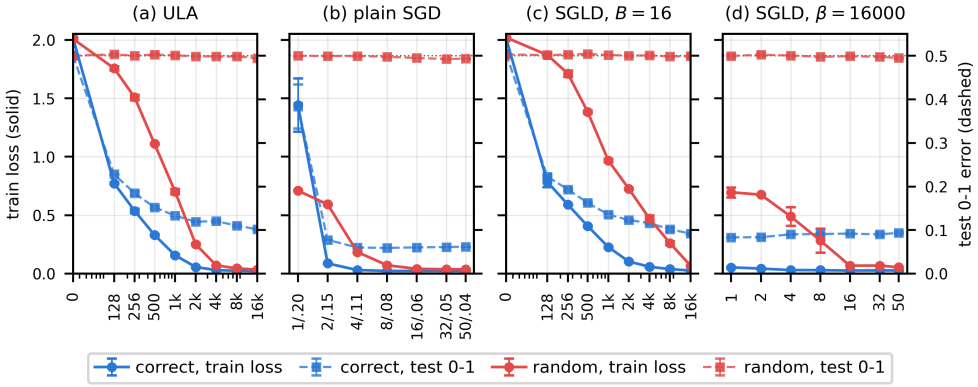


Figure 2: MNIST binary, BBCE, correct (blue) vs. random (red) labels, mean ± s.e.m. over 3 seeds (legend in figure). Panel x-axes: (a),(c) inverse temperature $\beta \in \{0, \ldots, 16000\}$; (b) plain SGD's own batch/step-size ladder; (d) batch size $B \in \{1, \ldots, 50\}$.

inverse-temperature parameter. The solid curves report the empirical training loss optimized by the algorithm (left axis); the dashed curves report test 0–1 error (right axis). Across the panels, the random-label models retain larger training loss on the hot part of the path, whereas both label conditions approach very small training loss at the cold end. The test curves remain separated even when the cold training losses are nearly indistinguishable: the early decay of training loss along the inverse-randomness axis is predictive of the final test error in the interpolation regime.

We used the ratio B/η rather than B alone for inverse-randomness parameter, which agrees with He et al. [2019], whose PAC-Bayes bound for SGD increases with exactly this ratio. Exact grids and other experimental details are in Appendix G, along with the BCE version (Figure 4); Figure 5 provides a better visualization of the area of the panels (a) and (c) in Figure 2.

CIFAR-100: hot training error predicts cold test error across tasks. Figure 1 shows the effect and Table 1 quantifies it, for two cold targets: full-batch ULA and full-batch GD, the deterministic limit of the plain-SGD family. Table 3 in Appendix H.2 shows every predictor against every target. Three things stand out. First, the cold model's own training error carries almost no signal: it interpolates on every task with a negative leave-one-out R^2 for both targets of Table 1. A hot predictor is not exact either, but cuts the error by a factor of 2.5 to 3 and its Spearman reaches 0.89 to 0.96 on the same tasks.

Second, a single hot training error is almost as good as the average over the hot path, which Equation 1 suggests. That is good news for efficiency: the J=1 case is the cheap method one would actually run.

Third, the best predictor of an algorithm's cold behavior is not its *own* hot training error. A cheap mini-batch SGLD run predicts both cold targets as accurately and for far less wall-clock (Table 1). Figure 10 in Appendix H.2 puts every predictor of every target on accuracy-against-cost axes.

References

Martin Anthony and Peter Bartlett. *Learning in Neural Networks: Theoretical Foundations*. Cambridge University Press, 1999.

Tzuu-Shuh Chiang, Chii-Ruey Hwang, and Shuenn Jyi Sheu. Diffusion for global optimization in Rˆn. *SIAM Journal on Control and Optimization*, 25(3):737–753, 1987.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent PAC-Bayes priors via differential privacy. *Advances in Neural Information Processing Systems*, 31, 2018.

Saul B Gelfand and Sanjoy K Mitter. Recursive stochastic algorithms for global optimization in rˆd. *SIAM Journal on Control and Optimization*, 29(5):999–1018, 1991.

Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems*, volume 32, pages 1141–1150, 2019.

Richard Holley and Daniel Stroock. Logarithmic sobolev inequalities and stochastic Ising models. *Journal of Statistical Physics*, 46(5-6):1159–1194, 1987.

Yuri Kinoshita and Taiji Suzuki. Improved convergence rate of stochastic gradient Langevin dynamics with variance reduction and its application to optimization. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Paul Langevin. Sur la théorie du mouvement brownien. *CR Acad. Sci. Paris*, 146(530-533):530, 1908.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

Andreas Maurer. A note on the PAC Bayesian theorem. *arXiv preprint cs/0411099*, 2004.

Andreas Maurer, Erfan Mirzaei, and Massimiliano Pontil. Generalization of Gibbs and Langevin Monte Carlo algorithms in the interpolation regime. *arXiv preprint arXiv:2510.06028*, 2025.

Giorgio Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703. PMLR, 2017.

Omar Rivasplata, Ilja Kuzborskij, Csaba Szepesvári, and John Shawe-Taylor. PAC-Bayes analysis beyond the usual bounds. *Advances in Neural Information Processing Systems*, 33:16833–16845, 2020.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Ilya O Tolstikhin and Yevgeny Seldin. PAC-Bayes-empirical-Bernstein inequality. *Advances in Neural Information Processing Systems*, 26, 2013.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. *Advances in Neural Information Processing Systems*, 32, 2019.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

In appendices, we first give a glossary of the notation used throughout the paper, then state the additional notations and assumptions, review the PAC-Bayesian results we use, give the proofs missing from the main text, and provide additional details and results for the numerical experiments.

A Table of notation

Notation	Brief description	Section
$\mathcal{X}$	space of data	2, B
Σ	sigma algebra (events) on $\mathcal{X}$	2, C
μ	probability distribution of the data on $\mathcal{X}$	2
n	sample size	2, 3
$\mathbf{x}$	training sample $(x_1, \ldots, x_n) \in \mathcal{X}^n$, drawn as $\mathbf{x} \sim \mu^n$	2, C
δ	confidence parameter in high probability bounds	2, C
$\mathcal{H}$	hypothesis space, also called the space of weights	2, B
Ω	sigma algebra (events) on $\mathcal{H}$	2, B
h	a hypothesis, $h \in \mathcal{H}$	2
d	dimension of the weight space, when $\mathcal{H} \subseteq \mathbb{R}^d$	B
ℓ	loss function $\ell : \mathcal{H} \times \mathcal{X} \to [0, \infty)$	2, 3
$L(h)$	$L(h) = \mathbb{E}_{x\sim\mu}[\ell(h, x)]$, expected (true) error of h	2, C
$\hat{L}(h, \mathbf{x})$	$\hat{L}(h, \mathbf{x}) = \frac{1}{n}\sum_i \ell(h, x_i)$, empirical error of h	2, D, E
$L_{\mathbf{01}},\ \hat{L}_{\mathbf{01}}$	expected and empirical 01-loss, the reported test error	C.2, 3
$\mathcal{P}(\mathcal{H})$	probability measures on $(\mathcal{H}, \Omega)$	2
$\mathcal{P}_\pi(\mathcal{H})$	$\{\nu : \mathrm{KL}(\nu, \pi) < \infty,\ \mathbb{E}_\nu[\hat{L}] \leq \mathbb{E}_\pi[\hat{L}]\}$	2
π	data-independent prior on $\mathcal{H}$	2, B
ν	stochastic algorithm, a map $\nu : \mathcal{X}^n \to \mathcal{P}(\mathcal{H})$	2, C
$\nu(r)$	one-parameter family of stationary distributions of an iterative algorithm	1, 2
r	inverse-randomness parameter	1, 2
s	a hotter value of the inverse-randomness parameter, $s < r$	1, 2
$\mathbb{E}_\nu\left[\hat{L}\right], \mathbb{V}_\nu\left[\hat{L}\right]$	mean and variance of the empirical error under $h \sim \nu(\mathbf{x})$	2, D
$\mathrm{KL}(\rho, \nu)$	$\mathbb{E}_{h\sim\rho}\left[\ln \frac{d\rho}{d\nu}(h)\right]$, relative entropy of $\rho, \nu \in \mathcal{P}(\mathcal{H})$	2, B
β	inverse temperature	1, 2, 3
$G(\beta)$	Gibbs distribution at inverse temperature β	2, E
G	the Gibbs family, $\beta \mapsto G(\beta)$	2
$Z(\beta)$	partition function $\int_{\mathcal{H}} e^{-\beta\hat{L}(h)}\, d\pi(h)$	2, E
N	$\{g : \hat{L}(g) = 0\}$, the set of empirical risk minimizers	E
$G(\infty)$	weak limit of $G(\beta)$, the prior conditioned on N	E
φ	$\varphi(\beta) = \mathbb{E}_{G(\beta)}[\hat{L}]$, training error along the Gibbs family	2, D
ψ	$\psi(s) = \mathbb{E}_{\nu(s)}[\hat{L}]$, training error along the alg. path	2, D
α	projecting the path to the Gibbs family, $\alpha = \varphi^{-1} \circ \psi$	2, D
$\Delta(r)$	$\min_{\beta\geq 0} \mathrm{KL}(\nu(r), G(\beta))$	2, D
$F(h, \mathbf{x})$	placeholder for the generalization gap	C
$\kappa(p, q)$	$p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}$, relative entropy of two Bernoulli variables	2, C.2
$\kappa^{-1}(p, t)$	$\inf\{q \geq p : \kappa(p, q) \geq t\}$, the direct inversion of κ	C.2
σ_{prior}	standard deviation (width) of the Gaussian prior π	3, G
η	step size or learning rate	3, G
B	mini-batch size	3, G

Notation	Brief description	Section
B_0, η_0, c_B, c_η	base values and ratios of the coupled SGD ladder $B_k = B_0 c_B^k$, $\eta_k = \eta_0 c_\eta^k$	G
λ	weight decay coefficient of the SGD and BCE runs	G
R^2	coefficient of determination of the leave-one-out fit from hot predictor to cold test error	3

B Additional Notation and assumptions

The relative entropy or Kullback-Leibler divergence between two probability measures is the function $\mathrm{KL} : (\rho, \nu) \in \mathcal{P}(\mathcal{H}) \times \mathcal{P}(\mathcal{H}) \mapsto \mathbb{E}_{h\sim\rho}\big[\ln \frac{d\rho}{d\nu}(h)\big]$ if ρ is absolutely continuous w.r.t. ν, otherwise the value is ∞. Although it is not symmetric nor satisfies the triangle inequality, we will loosely refer to the KL-divergence as a distance, just as the distance from a valley to a summit is asymmetric to the mountaineer.

B.1 Standing assumptions on the prior and the empirical error

These hold everywhere in the paper. The training sample $\mathbf{x}$ is fixed, so $\hat{L}$ is treated as a fixed function on $\mathcal{H}$ and we write $\hat{L}(h)$ for $\hat{L}(h, \mathbf{x})$.

- (A1) *Measurable spaces.* $(\mathcal{X}, \Sigma)$ is a measurable space of data with distribution μ, $(\mathcal{H}, \Omega)$ is a measurable space of hypotheses, and $\pi \in \mathcal{P}(\mathcal{H})$ is a prior that does not depend on the data.
- (A2) *Measurability.* $\hat{L} : \mathcal{H} \to [0, \infty)$ is measurable.
- (A3) *Nontriviality.* $0 < \mathbb{E}_\pi[\hat{L}]$ and the π-essential infimum of $\hat{L}$ is 0.
- (A4) *Moments.* $\mathbb{E}_\pi[\hat{L}] < \infty$ and $\mathbb{E}_\pi[\hat{L}^2] < \infty$.
- (A5) *Null level sets.* $\pi\{h : \hat{L}(h) = t\} = 0$ for every $t \in \mathbb{R}$: no level set of $\hat{L}$ carries prior mass, so $\hat{L}$ is non-constant π-almost surely and has no flat piece the prior can see.

(A3) and (A4) are what make φ a diffeomorphism of $[0, \infty)$ onto $(0, \mathbb{E}_\pi[\hat{L}]]$ in Proposition 8. If π is absolutely continuous w.r.t. Lebesgue measure on $\mathbb{R}^d$, then (A5) guarantees π-a.s. non-vanishing gradients and existence of the Langevin dynamics used to sample from π.

B.2 Assumptions on the family of stationary distributions

Section 2 studies a one-parameter family $\nu : [0, \infty) \to \mathcal{P}(\mathcal{H})$ of limiting distributions of an iterative algorithm. We assume:

- (F1) *Admissibility.* $\mathrm{KL}(\nu(r), \pi) < \infty$ and $\mathbb{E}_{\nu(r)}[\hat{L}] \le \mathbb{E}_\pi[\hat{L}]$ for every $r \ge 0$.
- (F2) *Hot end.* $\nu(0) = \pi$. The most random member of the path is the prior, which the path shares with the Gibbs family.
- (F3) *Monotonicity.* $\psi(r) = \mathbb{E}_{\nu(r)}[\hat{L}]$ is strictly decreasing. Less injected randomness gives a smaller training error. This is what makes $\alpha = \varphi^{-1} \circ \psi$ a bijection.

B.3 Extra hypotheses of individual results

Every result rests on (A1)–(A5), and every result about the family $\nu(\cdot)$ also rests on (F1)–(F3). The table lists what each one needs *in addition* to those.

Result	Additional hypotheses
Theorem 1	ℓ has values in $[0,1]$; $\nu(\mathbf{x}) \ll \pi$ for every $\mathbf{x} \in \mathcal{X}^n$; $n \geq 8$, which is what gives the $2\sqrt{n}$ term [Maurer, 2004]
Theorem 4	F measurable; $\nu(\mathbf{x}) \ll \pi$ for every $\mathbf{x}$
Bounds equation 7	the secondary loss used inside κ has values in $[0,1]$ and satisfies the conditions of Theorem 1 of Maurer [2004]
Corollary 5	$x \mapsto f(h,x)$ is σ-sub-Gaussian under μ for every $h \in \mathcal{H}$
Theorem 2 (v)	$\hat{L}$ has values in $[0,1]$; ψ continuously differentiable; $\mathbb{V}_{\nu(s)}[\hat{L}] > \sqrt{\Delta(s)/2}$ for all $s \leq r$
Proposition 12	$G(\beta)$ satisfies the log-Sobolev inequality equation 12 with constant $c_{\mathrm{LS}} > 0$; $-RI \preceq \nabla^2 \hat{L} \preceq RI$; step size $\eta \leq c_{\mathrm{LS}}/(4(\beta R + \sigma^{-2})^2)$
Prediction across tasks (Section 3)	the residuals $\Delta_k(r)$ are comparable across the tasks k; ρ is taken uniform on $[0,r]$, a maximum-entropy choice

C PAC-Bayes

We review PAC-Bayesian theory. There is no claim to novelty.

C.1 Generalized PAC-Bayesian Theorem

Definition 3. *Given a stochastic algorithm ν we define a probability measure ρ_ν on $\mathcal{H} \times \mathcal{X}^n$ by*

$$\rho_\nu(A) = \mathbb{E}_{\mathbf{x}\sim\mu^n}\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[1_A(h,\mathbf{x})\right] \text{ for } A \in \Omega \otimes \Sigma^{\otimes n}. \tag{2}$$

Then, $\mathbb{E}_{(h,\mathbf{x})\sim\rho_\nu}\left[\phi(h,\mathbf{x})\right] = \mathbb{E}_{\mathbf{x}}\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[\phi(h,\mathbf{x})\right]$ for measurable $\phi : \mathcal{H} \times \mathcal{X}^n \to \mathbb{R}$. To draw the pair $(h,\mathbf{x})$ from ρ_ν we first draw the training sample $\mathbf{x}$, and then sample h from $\nu(\mathbf{x})$.

Theorem 4. *Let $F : \mathcal{H} \times \mathcal{X}^n \to \mathbb{R}$ be some measurable function, and let ν be a stochastic algorithm such that $\nu(\mathbf{x})$ is absolutely continuous w.r.t. π for all$\mathbf{x} \in \mathcal{X}^n$. Then*

(i) for $\delta > 0$ with probability at least $1-\delta$ in $\mathbf{x} \sim \mu^n$ and $h \sim \nu(\mathbf{x})$

$$F(h,\mathbf{x}) \leq \ln \frac{d\nu(\mathbf{x})}{d\pi}(h) + \ln \mathbb{E}_{\mathbf{x}}\mathbb{E}_{g\sim\pi}\left[e^{F(g,\mathbf{x})}\right] + \ln\left(\frac{1}{\delta}\right)$$

(ii) for $\delta > 0$ with probability at least $1-\delta$ in $\mathbf{x} \sim \mu^n$

$$\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[F(h,\mathbf{x})\right] \leq \mathrm{KL}\left(\nu(\mathbf{x}),\pi\right) + \ln \frac{\mathbb{E}_{\mathbf{x}}\mathbb{E}_{g\sim\pi}\left[e^{F(g,\mathbf{x})}\right]}{\delta}$$

Proof. By Markov's inequality, for any real random variable Y

$$\Pr\left\{Y > \ln \mathbb{E}\left[e^Y\right] + \ln(1/\delta)\right\} = \Pr\left\{e^Y > \mathbb{E}\left[e^Y\right]/\delta\right\} \leq \delta.$$

To prove (i), we apply this to the random variable $Y = F(h,\mathbf{x}) - \ln\left(d\left(\nu(\mathbf{x})/d\pi\right)\right)(h)$ on the probability space $(\mathcal{H} \times \mathcal{X}^n, \Omega \otimes \Sigma^{\otimes n}, \rho_\nu)$. This gives, with probability at least $1-\delta$ as $\mathbf{x} \sim \mu^n$ and $h \sim \nu(\mathbf{x})$),

$$\begin{aligned}
&F(h,\mathbf{x}) - \ln \frac{d\nu(\mathbf{x})}{d\pi}(h) \\
&\leq \ln \mathbb{E}_{\mathbf{x}}\mathbb{E}_{g\sim\nu(\mathbf{x})}\left[e^{F(g,\mathbf{x}) - \ln(d(\nu(\mathbf{x})/d\pi))(g)}\right] + \ln(1/\delta) \\
&= \ln \mathbb{E}_{\mathbf{x}}\mathbb{E}_{g\sim\pi}\left[e^{F(g,\mathbf{x}) - \ln(d(\nu(\mathbf{x})/d\pi))(g) + \ln(d(\nu(\mathbf{x})/d\pi))(g)}\right] + \ln(1/\delta) \\
&= \ln \mathbb{E}_{\mathbf{x}}\mathbb{E}_{g\sim\pi}\left[e^{F(g,\mathbf{x})}\right] + \ln(1/\delta).
\end{aligned}$$

For (ii) apply Markov's inequality on the probability space $(\mathcal{X}^n, \Sigma^{\otimes n}, \mu^n)$ to the random variable $Y = \mathbb{E}_{g\sim\nu(\mathbf{x})}\left[F(g,\mathbf{x})\right] - \mathrm{KL}\left(\nu(\mathbf{x}),\pi\right) = \mathbb{E}_{g\sim\nu(\mathbf{x})}\left[F(g,\mathbf{x}) - \ln\left(d\left(\nu(\mathbf{x})/d\pi\right)\right)(g)\right]$ instead. By Jensen's inequality

$$e^{\mathbb{E}_{g\sim\nu(\mathbf{x})}[F(g,\mathbf{x}) - \ln(d(\nu(\mathbf{x})/d\pi))(g)]} \leq \mathbb{E}_{g\sim\nu(\mathbf{x})}\left[e^{F(g,\mathbf{x}) - \ln(d(\nu(\mathbf{x})/d\pi))(g)}\right].$$

Then proceed as before. □

C.2 κ-PAC-Bayesian Bounds and the 0–1 error

Assume that ℓ has values in $[0,1]$. To derive Theorem 1, let $F(h,\mathbf{x}) = n\,\kappa\left(\hat{L}(h,\mathbf{x}), L(h)\right)$, where κ is the relative entropy of two Bernoulli variables with expectations p and q

$$\kappa(p,q) = p\ln\frac{p}{q} + (1-p)\ln\frac{1-p}{1-q}. \tag{3}$$

for which Tolstikhin and Seldin [2013] give the inversion rule $\kappa(p,q) \leq B \implies q - p \leq \sqrt{2pB} + 2B$. Tonelli's theorem and Theorem 1 of Maurer [2004] then give $\mathbb{E}_{\mathbf{x}}\mathbb{E}_{g\sim\pi}\left[e^{F(g,\mathbf{x})}\right] = \mathbb{E}_{g\sim\pi}\mathbb{E}_{\mathbf{x}}\left[e^{F(g,\mathbf{x})}\right] \leq 2\sqrt{n}$ for $n \geq 8$. Substitution in (i) of Theorem 4 and division by n then gives with high probability as $\mathbf{x}\sim\mu^n$ and $h\sim\nu(\mathbf{x})$

$$\kappa\left(\hat{L}(h,\mathbf{x}), L(h)\right) \leq \frac{1}{n}\left(\ln\frac{d\nu(\mathbf{x})}{d\pi} + \ln\left(\frac{2\sqrt{n}}{\delta}\right)\right), \tag{4}$$

and the inversion rule of Tolstikhin and Seldin [2013] gives

$$L(h) \leq \hat{L}(h,\mathbf{x}) + \sqrt{\frac{2\hat{L}(h,\mathbf{x})}{n}\left(\ln\frac{d\nu(\mathbf{x})}{d\pi} + \ln\frac{2\sqrt{n}}{\delta}\right)} + \frac{2}{n}\left(\ln\frac{d\nu(\mathbf{x})}{d\pi} + \ln\frac{2\sqrt{n}}{\delta}\right). \tag{5}$$

The derivation of the bound for $\mathbb{E}_{h\sim\nu(\mathbf{x})}[L(h)]$ from Theorem 4 (ii) is analogous, we just have to use $\kappa\left(\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[\hat{L}(h,\mathbf{x})\right], \mathbb{E}_{h\sim\nu(\mathbf{x})}[L(h)]\right) \leq \mathbb{E}_{h\sim\nu(\mathbf{x})}\kappa\left(\hat{L}(h,\mathbf{x}), L(h)\right)$ by the joint convexity of κ (see Cover [1999]). This gives

$$\kappa\left(\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[\hat{L}(h,\mathbf{x})\right], \mathbb{E}_{h\sim\nu(\mathbf{x})}[L(h)]\right) \leq \frac{1}{n}\left(\mathrm{KL}(\nu(\mathbf{x}),\pi) + \ln\left(\frac{2\sqrt{n}}{\delta}\right)\right) \tag{6}$$

and the corresponding inequality obtained from the inversion rule.

The inversion rule produces directly interpretable bounds, but stronger is the direct inversion using the function $\kappa^{-1} : [0,1]\times[0,\infty)$ by

$$\kappa^{-1}(p,t) = \inf\{q : q \geq p, \kappa(p,q) \geq t\}.$$

In the definition of F, using the function κ, we can use other loss functions, possibly different from the loss function ℓ that defines the Gibbs posterior. If these loss functions satisfy the conditions of Theorem 1 in Maurer [2004], we obtain analogous bounds. In particular, for binary classification, we can use the 01 loss. Momentarily changing notation by replacing $x\in\mathcal{X}$ by (x,y), where y is the label corresponding to x, the 01 loss is defined as

$$\hat{L}_{\mathbf{01}}(h,\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} 1_{(-\infty,0)}\left(y_i\ell(h,x_i)\right)$$

and $L_{\mathbf{01}}(h) = \mathbb{E}_{x\sim\mu^n}\left[\hat{L}_{\mathbf{01}}(h,\mathbf{x})\right]$. We then obtain the bounds

$$\begin{aligned} L_{\mathbf{01}}(h) &\leq \kappa^{-1}\left(\hat{L}_{\mathbf{01}}(h,\mathbf{x}), \frac{1}{n}\left(\ln\frac{d\nu(\mathbf{x})}{d\pi} + \ln\left(\frac{2\sqrt{n}}{\delta}\right)\right)\right) \\ \mathbb{E}_{h\sim\nu(\mathbf{x})}[L_{\mathbf{01}}(h)] &\leq \kappa^{-1}\left(\mathbb{E}_{h\sim\nu(\mathbf{x})}\left[\hat{L}_{\mathbf{01}}(h,\mathbf{x})\right], \frac{1}{n}\left(\mathrm{KL}(\nu(\mathbf{x}),\pi) + \ln\left(\frac{2\sqrt{n}}{\delta}\right)\right)\right). \end{aligned} \tag{7}$$

C.3 Sub-gaussian losses

The freedom in the choice of F allows a number of bounds to be derived from Theorem 4. A centered real random variable Y is called σ-sub-Gaussian if $\ln\mathbb{E}e^{\lambda Y - \mathbb{E}Y} \leq \lambda^2\sigma^2/2$ for all $\lambda\in\mathbb{R}$. Now, suppose that for some real function f all the $x\in\mathcal{X}\mapsto f(h,x)$ are σ-sub-Gaussian as $x\sim\mu$. Let $\hat{L}(h,\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f(h,x_i)$ and $L(h) = E_x[f(h,\mathbf{x})]$. Then, $\mathbf{x}\in\mathcal{X}^n \mapsto \hat{L}(h,\mathbf{x})$ as $\mathbf{x}\sim\mu^n$ is $\sigma/\sqrt{n}$-sub-Gaussian. It is tempting to set $F(h,\mathbf{x}) = \lambda\left(L(h) - \hat{L}(h,\mathbf{x})\right)$ in Theorem 4, divide

by λ and then optimize over λ. Unfortunately, the last step is impossible, since the optimal λ is data-dependent in its dependence on Γ and ruins the exponential moment bound on F. A more careful argument establishes the following.

Here $\boldsymbol{\beta} = (\beta_1 < \cdots < \beta_K)$ is an increasing sequence of inverse temperatures with $\beta_0 := 0$, and Γ is the corresponding estimate of the log-density $\ln\left(dG(\beta_K)/d\pi\right)$ obtained by evaluating the integral of Lemma 10(v) on the grid $\boldsymbol{\beta}$,

$$\Gamma(h, \mathbf{x}, \boldsymbol{\beta}) = -\beta_K \hat{L}(h, \mathbf{x}) + \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \mathbb{E}_{g \sim G(\beta_{k-1})}\left[\hat{L}(g, \mathbf{x})\right]. \tag{8}$$

Corollary 5. *Suppose that for all $h \in \mathcal{H}$ the random variables $x \in \mathcal{X} \mapsto f(h, x)$ as $x \sim \mu$ are σ-sub-Gaussian. For $\delta > 0$ with probability at least $1 - \delta$ as $\mathbf{x} \sim \mu^n$ and $h \sim G(\beta_K)$, if $\Gamma(h, \mathbf{x}, \boldsymbol{\beta}) \geq 1$ for Γ as in equation 8, then*

$$E_{x \sim \mu}[f(h, x)] - \frac{1}{n} \sum_i f(h, x_i) \leq \sigma \sqrt{\frac{2[\Gamma(h, \mathbf{x}, \boldsymbol{\beta})(1 + 1/n) + \ln(\Gamma(h, \mathbf{x}, \boldsymbol{\beta})(n+1)/\delta)]}{n}}.$$

For the proof, we use the following auxiliary result.

Lemma 6. *[Anthony and Bartlett, 1999, Lemma 15.6] Suppose* Pr *is a probability distribution and*

$$\{E(\alpha_1, \alpha_2, \delta) : 0 < \alpha_1, \alpha_2, \delta \leq 1\}$$

is a set of events, such that

(i) For all $0 < \alpha \leq 1$ and $0 < \delta \leq 1$,

$$\Pr\{E(\alpha, \alpha, \delta)\} \leq \delta.$$

(ii) For all $0 < \alpha_1 \leq \alpha \leq \alpha_2 \leq 1$ and $0 < \delta_1 \leq \delta \leq 1$

$$E(\alpha_1, \alpha_2, \delta_1) \subseteq E(\alpha, \alpha, \delta).$$

Then for $0 < a, \delta < 1$,

$$\Pr \bigcup_{\alpha \in (0,1]} E(\alpha a, \alpha, \delta\alpha(1 - a)) \leq \delta.$$

We put this lemma in a more convenient form.

Lemma 7. *Let Y and $X \geq 0$ be real random variables, $\psi : \mathbb{R} \times (0, 1) \to \mathbb{R}$ be increasing in the 1st argument and $\forall C > 1$, $\delta \in (0, 1)$,*

$$\Pr\{X \leq C \wedge Y > \psi(C, \delta)\} < \delta.$$

Then, for every $\epsilon > 0$

$$\Pr\left\{X \geq 1 \wedge Y > \psi\left(X(1 + \epsilon), \frac{\delta\epsilon}{X(1 + \epsilon)}\right)\right\} < \delta.$$

Proof. This follows from Lemma 6 using the events

$$E(\alpha_1, \alpha_2, \delta) = \left\{X \leq \alpha_2^{-1} \wedge Y > \psi\left(\alpha_1^{-1}, \delta\right)\right\}$$

and $a = 1/(1 + \epsilon)$. □

Proof of Corollary 5. Take $F = \lambda\left(L(h) - \hat{L}(h, \mathbf{x})\right)$, so $\ln \mathbb{E}_{\mathbf{x}} \mathbb{E}_{g \sim \pi}\left[e^{F(g, \mathbf{x})}\right] \leq \lambda^2 \sigma^2 / (2n)$. From Theorem 4 and the properties of sub-Gaussian variables we get with $\lambda = \sigma^{-1}\sqrt{2n(C + \ln(1/\delta))}$ that

$$\begin{aligned}
& \Pr\left\{\Gamma(h, \mathbf{x}, \boldsymbol{\beta}) \leq C \wedge L(h) - \hat{L}(h, \mathbf{x}) > \sigma\sqrt{\frac{2(C + \ln(1/\delta))}{n}}\right\} \\
= \; & \Pr\left\{\Gamma(h, \mathbf{x}, \boldsymbol{\beta}) \leq C \wedge L(h) - \hat{L}(h, \mathbf{x}) > \frac{C + \ln(1/\delta)}{\lambda} + \frac{\lambda\sigma^2}{2n}\right\} \\
= \; & \Pr\left\{\Gamma(h, \mathbf{x}, \boldsymbol{\beta}) \leq C \wedge \lambda\left(L(h) - \hat{L}(h, \mathbf{x})\right) > C + \lambda^2\sigma^2/(2n) + \ln(1/\delta)\right\} \leq \delta.
\end{aligned}$$

Substitution in Lemma 7 with $\psi(C, \delta) = \sigma\sqrt{2(C + \ln(1/\delta))/n}$, $X = \Gamma(h, \mathbf{x}, \boldsymbol{\beta})$, $Y = L(h) - \hat{L}(h, \mathbf{x})$ and $\epsilon = 1/n$ gives Corollary 5. □

D Proofs for Section 2

The following lists the properties of the Gibbs family which are most important for our purposes. The relevant lemmata are in Appendix E.

Proposition 8. *(i) The function* $\varphi : \beta \mapsto \mathbb{E}_{G(\beta)}\left[\hat{L}\right]$ *is differentiable and strictly decreasing in* β *with derivative* $-\mathbb{V}_{G(\beta)}\left[\hat{L}\right]$*. It is a diffeomorphism from* $[0,\infty)$ *onto* $\left(0, \mathbb{E}_\pi\left[\hat{L}\right]\right]$

(ii) $\mathrm{KL}\left(G\left(\beta\right),\pi\right) = \int_0^\beta \left(\varphi\left(\gamma\right) - \varphi\left(\beta\right)\right) d\gamma$*.*

(iii) The function $\beta \mapsto \mathrm{KL}\left(G\left(\beta\right),\pi\right)$ *is differentiable and strictly increasing with derivative* $\beta\, \mathbb{V}_{G_\beta}\left[\hat{L}\right]$*. It is a diffeomorphism from* $[0,\infty)$ *onto* $[0,\infty)$*.*

Proof. (i) follows from Lemma 10 (iii) and Lemma 11(i).

(ii) is Lemma 10 (vi).

(iii) follows from Lemma 11 (iii) and our assumption $\pi(N) = 0$. □

We restate and prove Theorem 2

Theorem 9. *There exists a bijective function* $\alpha : [0,\infty) \to [0,\infty)$*, such that*

(i) $\Delta\left(r\right) = \mathrm{KL}\left(\nu\left(r\right), G\left(\alpha\left(r\right)\right)\right)$

(ii) $\psi\left(r\right) = \varphi\left(\alpha\left(r\right)\right)$

(iii) $\mathrm{KL}\left(\nu\left(r\right),\pi\right) = \Delta\left(r\right) + \mathrm{KL}\left(G\left(\alpha\left(r\right)\right),\pi\right) = \Delta\left(r\right) + \int_0^{\alpha(r)} \left(\varphi\left(\gamma\right) - \psi\left(r\right)\right) d\gamma$

(iv) there is a probability measure ρ *on the interval* $[0,r]$ *such that*

$$\mathrm{KL}\left(\nu\left(r\right),\pi\right) = \Delta\left(r\right) + \alpha\left(r\right)\left(\mathbb{E}_{s\sim\rho}\left[\psi\left(s\right)\right] - \psi\left(r\right)\right). \tag{9}$$

(v) If $\hat{L}$ *has values in* $[0,1]$*, if* ψ *is also continuously differentiable, and* $\mathbb{V}_{\nu(s)}\left[\hat{L}\right] > \sqrt{\Delta\left(s\right)/2}$ *for all* $s \le r$*, then*

$$\mathrm{KL}\left(\nu\left(r\right),\pi\right) \le \Delta\left(r\right) + \int_0^r \left(\psi\left(s\right) - \psi\left(r\right)\right) \frac{\left(-\psi'\left(s\right)\right)}{\mathbb{V}_{\nu(s)}\left[\hat{L}\right] - \sqrt{\Delta\left(s\right)/2}} ds.$$

Before giving the proof we comment on the theorem. See also the illustration in Figure 3. By the PAC-Bayesian theorem the quantity to bound is the KL-divergence from $\nu\left(r\right)$ to the prior $\pi = \nu\left(0\right)$. By (i) and (iii) this decomposes as

$$KL\left(\nu\left(r\right),\pi\right) = KL\left(\nu\left(r\right), G\left(\alpha\left(r\right)\right)\right) + KL\left(G\left(\alpha\left(r\right)\right),\pi\right).$$

If the KL-divergence was the square of a Euclidean distance (which it isn't), then this would be the Pythagorean theorem. The first term on the right hand side is the residual error $\Delta\left(r\right)$, quantifying the distance from the Gibbs family. The second term has an integral representation as in [Maurer et al., 2025]. By (ii) the empirical errors of $G\left(\gamma\right)$ and $\nu\left(\alpha^{-1}\left(\gamma\right)\right)$ are the same, so the bijection α can pull the scaled Lebesgue measure back to the interval $[0,r]$, as expressed in part (iv). Part (v) finally gives an explicit bound, with the residual Δ as the only unobservable quantity.

Proof. Since $\nu\left(r\right) \in \mathcal{P}_\pi$ we have $\psi\left(s\right) = \mathbb{E}_\nu\left[\hat{L}\right] \in \left(0, \mathbb{E}_\pi\left[\hat{L}\right]\right]$. By Proposition 8 (i) the function φ is a bijection of $[0,\infty)$ to $\left(0, \mathbb{E}_\pi\left[\hat{L}\right]\right]$, so $\alpha\left(r\right) := \varphi^{-1}\left(\psi\left(r\right)\right)$ is well defined and satisfies (ii). Since ψ is strictly decreasing (F3), α is injective, and by (F2) $\psi\left(0\right) = \mathbb{E}_\pi[\hat{L}]$, so $\alpha\left(0\right) = 0$. It is onto $[0,\infty)$, and hence a bijection, whenever $\psi\left(r\right) \to 0$ as $r \to \infty$; parts (i)–(v) below use only injectivity on $[0,r]$.

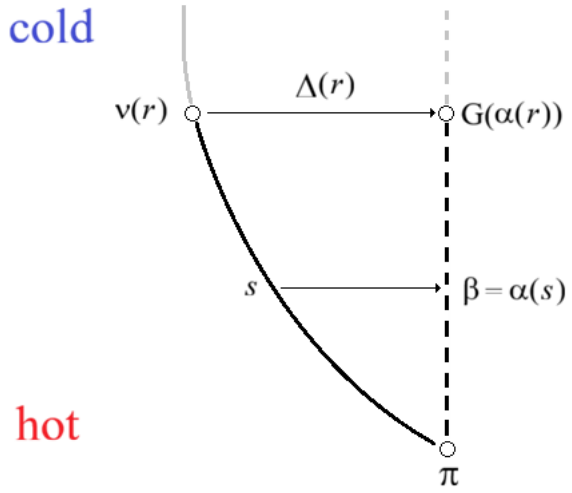


Figure 3: An illustration of Theorem 2. $\nu(r)$ is projected to the nearest (in relative entropy) member $G(\alpha(r))$ of the Gibbs family.

(i) Fix r and for $\beta \geq 0$ let $f(\beta) = \mathrm{KL}(\nu(r), G(\beta))$, so

$$\begin{aligned} f(\beta) &= \mathbb{E}_{\nu(r)}\left[\ln\left(\frac{d\nu(r)}{d\pi}\frac{d\pi}{dG(\beta)}\right)\right] \\ &= \mathrm{KL}(\nu(r), \pi) + \beta\mathbb{E}_{\nu(r)}\left[\hat{L}\right] + \ln Z(\beta) \\ &= \mathrm{KL}(\nu(r), \pi) + \beta\psi(r) - \int_0^\beta \varphi(\gamma)\, d\gamma. \end{aligned}$$

Since $\mathrm{KL}(\nu(r), \pi) < \infty$ the function f is well defined and twice differentiable in β with derivatives

$$\begin{aligned} f'(\beta) &= \psi(r) - \varphi(\beta) \\ f''(\beta) &= \mathbb{V}_{G(\beta)}\left[\hat{L}\right]. \end{aligned}$$

This shows that f is strictly convex, and that $\beta = \alpha(r)$ is the unique minimizer, so $\Delta(r) = \mathrm{KL}(\nu(r), G(\alpha(r)))$.

(iii) Decomposition of the KL-divergence gives

$$\begin{aligned} \mathrm{KL}(\nu(r), \pi) &= \mathrm{KL}(\nu(r), G(\alpha(r))) + \mathbb{E}_{\nu(r)}\left[\ln\frac{dG(\alpha(r))}{d\pi}\right] \\ &= \mathrm{KL}(\nu(r), G(\alpha(r))) + \int_0^{\alpha(r)} (\varphi(\gamma) - \psi(r))\, d\gamma \qquad (10) \\ &= \mathrm{KL}(\nu(r), G(\alpha(r))) + \mathrm{KL}(G(\alpha(r)), \pi). \end{aligned}$$

In the last step we used (ii) and the integral formula Proposition 8(ii) for the KL-divergence of Gibbs distributions to the prior.

(iv) Since $\nu(0) = \pi$ we have $\alpha(0) = 0$. Denote Lebesgue measure on $\mathbb{R}$ by m. By (iii) and the definition of α we have

$$\begin{aligned} \mathrm{KL}(\nu(r), \pi) &= \Delta(r) + \mathrm{KL}(G(\alpha(r)), \pi) \\ &= \Delta(r) + \int_0^{\alpha(r)} \varphi(\gamma)\, dm(\gamma) - \alpha(r)\varphi(\alpha(r)). \\ &= \Delta(r) + \alpha(r)\left(\int_0^{\alpha(r)} \varphi(\gamma)\frac{dm(\gamma)}{\alpha(r)} - \psi(r)\right). \end{aligned}$$

The bijection α^{-1} maps the interval $[0, \alpha(r)]$ onto $[0, r]$. Then the measure defined by $\rho(A) = m(\alpha(A))/\alpha(r)$ for measurable $A \subseteq [0, r]$ is a probability measure on $[0, r]$ and

$$\int_0^{\alpha(r)} \varphi(\gamma) \frac{dm(\gamma)}{\alpha(r)} = \int_0^{\alpha(r)} \psi\left(\alpha^{-1}(\gamma)\right) \frac{dm(\gamma)}{\alpha(r)} = \mathbb{E}_{s\sim\rho}[\psi(s)].$$

(v) The change of variables formula gives

$$\int_0^{\alpha(r)} \varphi(\gamma) d\gamma = \int_0^r \varphi(\alpha(s)) \alpha'(s) ds = \int_0^r \psi(s) \alpha'(s) ds.$$

Also we have $\alpha(t) = \int_0^t \alpha'(s) ds$, so

$$\begin{aligned} \mathrm{KL}(G(\alpha(r)), \pi) &= \int_0^r (\psi(s) - \psi(r)) \alpha'(s) ds \\ &= \int_0^r (\psi(s) - \psi(r)) \frac{-\psi'(s)}{\mathbb{V}_{G(\alpha(s))}\left[\hat{L}\right]} ds, \end{aligned}$$

where we used the formula $\varphi'(\beta) = -\mathbb{V}_{G(\beta)}\left[\hat{L}\right]$ (Lemma 10 (iii)), the inverse function theorem and the chain rule for the expression of $\alpha'(s)$. But since $\hat{L} \in [0, 1]$, we get $\mathbb{V}_{G(\alpha(s))} \geq \mathbb{V}_{\nu(s)} - \sqrt{\Delta(s)/2}$ from Pinsker's inequality. □

E General properties of the Gibbs distribution

Lemma 10. *(i) The Gibbs posterior $G(\beta)$ exists in $\mathcal{P}(\mathcal{H})$. $Z(0) = 1$ and $0 < Z(\beta) < 1$ for $\beta > 0$.*

(ii) $\varphi(0) = \mathbb{E}_\pi\left[\hat{L}\right]$, $\varphi(\beta) = \mathbb{E}_{G(\beta)}\left[\hat{L}\right] < \infty$, $\mathbb{E}_{G(\beta)}\left[\hat{L}^2\right] < \infty$ and $0 < \mathbb{V}_{G(\beta)}\left[\hat{L}\right] < \infty$.

(iii) The function $\varphi : \beta \mapsto \mathbb{E}_{G(\beta)}\left[\hat{L}\right]$ is differentiable and strictly decreasing in β with derivative $-\mathbb{V}_{G(\beta)}\left[\hat{L}\right]$.

(iv) $-\ln Z(\beta) = \int_0^\beta \varphi(\gamma) d\gamma$.

(v) $\ln \frac{dG(\beta)}{d\pi}(h) = -\beta \hat{L}(h) + \int_0^\beta \varphi(\gamma) d\gamma$.

(vi) $\mathrm{KL}(G(\beta), \pi) = -\beta \varphi(\beta) + \int_0^\beta \varphi(\gamma) d\gamma$.

(vii) The function $\beta \mapsto \mathrm{KL}(G(\beta), \pi)$ is differentiable with derivative $\beta\, \mathbb{V}_{G(\beta)}\left[\hat{L}\right]$, and strictly increasing for $\beta > 0$.

Proof. (i) By nontriviality there is $\epsilon > 0$ such that $0 < \pi\left\{\hat{L} < \epsilon\right\} < 1$. Note that $Z(0) = 1$ and for $\beta > 0$

$$\begin{aligned} 0 &< e^{-\beta\epsilon} \pi\left\{\hat{L} < \epsilon\right\} \leq \int_{\left\{\hat{L}<\epsilon\right\}} e^{-\beta\hat{L}} d\pi \\ &\leq Z(\beta) = \int_{\left\{\hat{L}<\epsilon\right\}} e^{-\beta\hat{L}} d\pi + \int_{\left\{\hat{L}\geq\epsilon\right\}} e^{-\beta\hat{L}} d\pi \\ &\leq \pi\left\{\hat{L} < \epsilon\right\} + e^{-\beta\epsilon} \pi\left\{\hat{L} \geq \epsilon\right\} < 1 = Z(0), \end{aligned}$$

so $0 < Z(\beta) < 1$. Since $0 < Z(\beta)$ the Gibbs posterior exists. Elementary verifications show that it is a probability measure.

(ii) By the moment assumptions for $\beta \geq 0$ we have

$$\mathbb{E}_{G(\beta)}\left[\hat{L}\right] = \frac{\int \hat{L} e^{-\beta\hat{L}} d\pi}{\int e^{-\beta\hat{L}} d\pi} \leq \frac{\mathbb{E}_\pi\left[\hat{L}\right]}{Z(\beta)} < \infty,$$

and similarly $\mathbb{E}_{G(\beta)}\left[\hat{L}^2\right] < \infty$ and thus $\mathbb{V}_{G(\beta)}\left[\hat{L}\right] < \infty$. Fix any $\beta \geq 0$ and let $a \in \mathbb{R}$. By nontriviality $\pi\left\{\hat{L} \neq a\right\} > 0$, so $\int_{\left\{\hat{L}\neq a\right\}} e^{-\beta\hat{L}} d\pi > 0$, so $G(\beta)\left\{\hat{L} \neq a\right\} > 0$. Since this holds for any a, $\hat{L}$ cannot be constant a.e. w.r.t. $G(\beta)$, so $\mathbb{V}_{G(\beta)}\left[\hat{L}\right] > 0$.

(iii) Define the real function $A(\beta) = -\ln Z(\beta)$. Differentiation shows $A'(\beta) = \mathbb{E}_{G(\beta)}\left[\hat{L}\right] = \varphi(\beta)$ and $A''(\beta) = -\mathbb{V}_{G(\beta)}\left[\hat{L}\right]$. Therefore φ is continuous and differentiable in β, with derivative $-\mathbb{V}_{G(\beta)}\left[\hat{L}\right]$. Together with (ii) this also implies that φ is strictly decreasing in β.

(iv) Since for A as defined above $A(0) = 0$, the fundamental theorem of calculus gives

$$-\ln Z(\beta) = \int_0^\beta \varphi(\gamma)\, d\gamma.$$

(v) and (vi) follow directly from (iv), since $\ln \frac{dG(\beta)}{d\pi}(h) = -\beta\, \hat{L}(h) - \ln Z(\beta)$. In combination with (iii) this also gives (vii). □

The next lemma lists asymptotic properties. We denote with $N = \left\{g \in \mathcal{H} : \hat{L}(g) = 0\right\}$ the set of empirical risk minimizers. If $\pi(N) > 0$ denote with $G(\infty)$ the probability measure defined by

$$G(\infty)(A) = \frac{\pi(A \cap N)}{\pi(N)} \text{ for } A \in \Omega.$$

Lemma 11. *(i)* $\lim_{\beta\to\infty} \varphi(\beta) = 0$, $\varphi([0,\infty)) = \left(0, \mathbb{E}_\pi\left(\hat{L}\right)\right]$ *and* φ *is a diffeomorphism.*

(ii) If $\pi(N) > 0$ *then* $G(\beta) \to G_\infty$ *weakly.*

(iii) $\lim_{\beta\to\infty} \mathrm{KL}(G(\beta), \pi) = -\ln \pi(N)$*, where either both are finite or both are infinite.*

(iv) If $\{\mathrm{KL}(G(\beta), \pi) : \beta \geq 0\}$ *is bounded then* $\pi(N) > 0$ *and* $G(\beta) \to G_\infty$ *weakly.*

Proof. (i) For any $\epsilon > 0$

$$\begin{aligned}
\varphi(\beta) &= \mathbb{E}_{G(\beta)}\left[\hat{L}\right] \leq \epsilon + \mathbb{E}_{G(\beta)}\left[\hat{L}1_{\left\{\hat{L}>\epsilon\right\}}\right] = \epsilon + \frac{\int_{\left\{\hat{L}>\epsilon\right\}} \hat{L} e^{-\beta\hat{L}} d\pi}{\int_{\mathcal{H}} e^{-\beta\hat{L}} d\pi} \\
&\leq \epsilon + \frac{\int_{\left\{\hat{L}>\epsilon\right\}} \hat{L} e^{-\beta\hat{L}} d\pi}{\int_{\left\{\hat{L}<\epsilon/2\right\}} e^{-\beta\hat{L}} d\pi} \leq \epsilon + \frac{\int_{\left\{\hat{L}>\epsilon\right\}} \hat{L} e^{-\beta\epsilon} d\pi}{e^{-\beta\epsilon/2}\pi\left\{\hat{L} < \epsilon/2\right\}} \\
&= \epsilon + \frac{e^{-\beta\epsilon/2} \int_{\left\{\hat{L}>\epsilon\right\}} \hat{L} d\pi}{\pi\left\{\hat{L} < \epsilon/2\right\}} \to \epsilon
\end{aligned}$$

as $\beta \to \infty$, since the denominator of the second term is positive by definition of the essential infimum. Thus $\lim_{\beta\to\infty} \varphi(\beta) < \epsilon$, for every $\epsilon > 0$. This gives the first part. The other assertion then follows from Lemma 10 (ii) and (iii).

(ii) For bounded measurable ψ

$$\begin{aligned}
\mathbb{E}_{G(\beta)}[\psi] &= \frac{\int_N \psi e^{-\beta\hat{L}} d\pi + \int_{N^c} \psi e^{-\beta\hat{L}} d\pi}{\int_N e^{-\beta\hat{L}} d\pi + \int_{N^c} e^{-\beta\hat{L}} d\pi} \\
&= \frac{\int_N \psi d\pi + \int_{N^c} \psi e^{-\beta\hat{L}} d\pi}{\pi(N) + \int_{N^c} e^{-\beta\hat{L}} d\pi},
\end{aligned}$$

and by bounded convergence both second terms in denominator and numerator go to zero as $\beta \to \infty$. (iii) By dominated convergence

$$Z(\beta) = \int_{\mathcal{H}} e^{-\beta \hat{L}} d\pi \to \pi(N),$$

as $\beta \to \infty$. Now we define functions $f_\beta(\gamma)$ for $\beta, \gamma \in [0, \infty)$ by

$$f_\beta(\gamma) = \begin{cases} \mathbb{E}_{G(\gamma)}\left[\hat{L}\right] - \mathbb{E}_{G(\beta)}\left[\hat{L}\right] & \text{if } \gamma \leq \beta \\ 0 & \text{if } \gamma > \beta \end{cases}.$$

The f_β are a non-decreasing family of non-negative functions, which as $\beta \to \infty$ converge pointwise to f, where $f(\gamma) = \mathbb{E}_{G(\gamma)}\left[\hat{L}\right]$, since by part (i) $\mathbb{E}_{G(\beta)}\left[\hat{L}\right] \to 0$. It follows from the monotone convergence theorem, that

$$\lim_{\beta \to \infty} \int_0^\infty f_\beta(x)\, dx = \int_0^\infty f(x)\, dx,$$

which may be finite or infinite. By Lemma 10 (vi) $\mathrm{KL}(G(\beta), \pi) = \int_0^\infty f_\beta(x)\, dx$, so

$$\begin{aligned} \lim_{\beta \to \infty} \mathrm{KL}(G(\beta), \pi) &= \lim_{\beta \to \infty} \int_0^\infty f_\beta(x)\, dx = \lim_{\beta \to \infty} \int_0^\beta \mathbb{E}_{G(\gamma)}\left[\hat{L}\right] d\gamma \\ &= -\lim_{\beta \to \infty} \ln Z(\beta) = -\ln \pi\left\{\hat{L} = 0\right\}. \end{aligned}$$

(iv) just combines the conclusions of (ii) and (iii). □

F The algorithms we run

This paper is about iterative learning algorithms that follow the negative gradient of the empirical error and inject noise at every step. ULA, SGLD and plain SGD are the three we run, and they differ only in where that noise comes from: ULA adds Gaussian noise to the full gradient, SGD gets its noise from the choice of mini-batch, and SGLD has both. This appendix writes down each update rule, says what plays the role of the inverse-randomness parameter r, and reports what is known about the residual Δ, which is precisely what that difference in the source of the noise costs.

Throughout, π is the centered Gaussian prior of width σ on $\mathbb{R}^d$, so the Gibbs posterior $G(\beta)$ has density proportional to e^{-V_β} for the potential

$$V_\beta(h) \;=\; \beta \hat{L}(h) + \frac{\|h\|^2}{2\sigma^2}. \tag{11}$$

A log-Sobolev inequality. The convergence results we quote need one assumption on $G(\beta)$. A measure ν on $\mathbb{R}^d$ satisfies a log-Sobolev inequality (LSI) with constant $c_{\mathrm{LS}} > 0$ if for every smooth $f : \mathbb{R}^d \to \mathbb{R}$

$$\mathbb{E}_{h\sim\nu}\big[f^2(h) \ln f^2(h)\big] - \mathbb{E}_{h\sim\nu}\big[f^2(h)\big] \ln \mathbb{E}_{h\sim\nu}\big[f^2(h)\big] \;\leq\; \frac{2}{c_{\mathrm{LS}}}\, \mathbb{E}_{h\sim\nu}\big[\|\nabla f(h)\|^2\big]. \tag{12}$$

An LSI holds when V_β is strongly convex, and, more usefully here, it survives bounded perturbations [Holley and Stroock, 1987]: the Gaussian prior alone satisfies equation 12 with constant σ^{-2}, so if $\hat{L}$ has values in $[0, 1]$ then $G(\beta)$ satisfies equation 12 with $c_{\mathrm{LS}} \geq \sigma^{-2} e^{-\beta}$. This is enough for our bounded losses, but it also shows the constant degrades as the temperature falls. Raginsky et al. [2017] give an LSI constant independent of $\mathbf{x}$ under dissipativity conditions on the loss, and Vempala and Wibisono [2019] list further non-log-concave examples.

Continuous Langevin dynamics (CLD). The idealization of all three algorithms is the diffusion

$$dh_t \;=\; -\Big(\nabla \hat{L}(h_t) + \frac{h_t}{\beta\sigma^2}\Big)dt + \sqrt{2/\beta}\; dB_t, \tag{13}$$

with B_t a standard Brownian motion in $\mathbb{R}^d$. Its stationary law is exactly $G(\beta)$ [Chiang et al., 1987], and under equation 12 the law $\nu_{\beta,t}$ of h_t converges to it geometrically in relative entropy, $\mathrm{KL}(\nu_{\beta,t}, G(\beta)) \leq e^{-2c_{\mathrm{LS}} t/\beta}\, \mathrm{KL}(\nu_{\beta,0}, G(\beta))$, by the standard argument of Vempala and Wibisono [2019] applied to equation 11. So for CLD the family $\nu(\cdot)$ *is* the Gibbs family, $r = \beta$, α is the identity and $\Delta \equiv 0$. CLD cannot be run; it is the reference point against which the other two are measured.

The unadjusted Langevin algorithm (ULA). Discretizing equation 13 with the Euler method and step size η gives the algorithm we run at full batch,

$$h_{t+1} \;=\; h_t - \eta \nabla \hat{L}(h_t) - \frac{\eta h_t}{\beta \sigma^2} + \sqrt{\frac{2\eta}{\beta}}\,\xi_t, \qquad \xi_t \sim \mathcal{N}(0, I) \text{ independent.} \tag{14}$$

Discretization biases the stationary law: as $t \to \infty$ the law of h_t converges to some $\nu_{\beta,\eta}$ that is close to, but not equal to, $G(\beta)$. Theorem 2 of Vempala and Wibisono [2019] controls the gap, and specializes to equation 11 as follows.

Proposition 12 (ULA bias, after Vempala and Wibisono, 2019)**.** *Suppose $G(\beta)$ satisfies equation 12 with constant $c_{\mathrm{LS}} > 0$, that $-RI \preceq \nabla^2 \hat{L}(h) \preceq RI$ for all h and some $R < \infty$, and that $0 < \eta \le \beta c_{\mathrm{LS}} / \big(4(\beta R + \sigma^{-2})^2\big)$. Let $\nu_{\beta,\eta,t}$ be the law of h_t in equation 14 started from $\nu_{\beta,0}$, and let $D(\beta) = \mathrm{KL}(\nu_{\beta,0}, G(\beta))$. Then for every $t \ge 0$*

$$\mathrm{KL}\big(\nu_{\beta,\eta,t}, G(\beta)\big) \;\le\; e^{-c_{\mathrm{LS}} \eta t/\beta} D(\beta) + \frac{8\eta d}{\beta\, c_{\mathrm{LS}}} \Big(\beta R + \frac{1}{\sigma^2}\Big)^2.$$

This follows from Vempala and Wibisono [2019] with the substitutions $V = \beta \hat{L} + \|h\|^2/(2\sigma^2)$, step size $\epsilon = \eta/\beta$ and Hessian bound $\beta R + \sigma^{-2}$, under which their algorithm becomes equation 14 and their target becomes $G(\beta)$. The first term is the cost of starting in the wrong place and dies out; the second bounds how far the stationary law sits from the Gibbs posterior.

Proposition 12 bounds the residual of Section 2 directly. Taking $\nu(r) = \nu_{\beta,\eta}$ with $r = \beta$ at fixed η, and using that Δ is a minimum over the whole Gibbs family,

$$\Delta(\beta) \;=\; \min_{\gamma \ge 0} \mathrm{KL}\big(\nu_{\beta,\eta}, G(\gamma)\big) \;\le\; \mathrm{KL}\big(\nu_{\beta,\eta}, G(\beta)\big) \;\le\; \frac{8\eta d}{\beta\, c_{\mathrm{LS}}} \Big(\beta R + \frac{1}{\sigma^2}\Big)^2. \tag{15}$$

The bound is proportional to η, so ULA can be brought as close to the Gibbs family as one likes by paying with more steps. This is why we treat ULA as the algorithm with the smallest residual.

Stochastic gradient Langevin dynamics (SGLD). SGLD replaces the full gradient in equation 14 by an unbiased estimate $\nabla \hat{L}_B$ computed on a mini-batch of size B [Welling and Teh, 2011, Gelfand and Mitter, 1991]:

$$h_{t+1} \;=\; h_t - \eta \nabla \hat{L}_B(h_t) - \frac{\eta h_t}{\beta \sigma^2} + \sqrt{\frac{2\eta}{\beta}}\,\xi_t. \tag{16}$$

There are now two sources of randomness, the injected Gaussian noise and the choice of mini-batch, and two knobs that move along the path: the temperature β at fixed B, and the batch size B at fixed β. Figure 2 uses the first in panel (c) and the second in panel (d). At $B = n$ the algorithm is ULA, and the mini-batch contribution vanishes.

At a fixed step size the mini-batch noise pushes the stationary law of equation 16 further from $G(\beta)$ than discretization alone does, because that noise is neither isotropic nor of constant covariance. Its effect is of higher order in η than the injected noise, though, so as $\eta \to 0$ SGLD approaches the Gibbs family just as ULA does [Welling and Teh, 2011]. We expect Δ to be larger for SGLD than for ULA at the step sizes we actually use, not in the limit.

We do not give a bound of the form equation 15 for SGLD. Relative-entropy guarantees do exist in this area: Kinoshita and Suzuki [2022] prove KL bounds under a log-Sobolev inequality, but for variance-reduced variants rather than the plain mini-batch chain we run, and the standard non-asymptotic results for plain SGLD are in Wasserstein distance under dissipativity [Raginsky et al., 2017]. Adapting either to bound Δ for our chain is possible in principle and we have not done it, so we treat SGLD's residual as a quantity our experiments measure rather than one we bound.

Stochastic Gradient Descent (SGD). Removing the injected noise from equation 16 leaves

$$h_{t+1} \;=\; h_t - \eta \nabla \hat{L}_B(h_t) - \eta \lambda h_t, \tag{17}$$

with weight decay λ. Mini-batch selection is now the only randomness, and there is no β to vary. Under the constant-covariance approximation of Mandt et al. [2017], which treats the mini-batch gradient noise as Gaussian with fixed covariance, equation 17 behaves near a minimum like an

Ornstein–Uhlenbeck process whose stationary covariance scales with η/B. That is, the ratio B/η plays the role that β plays for the Langevin algorithms. He et al. [2019] reach the same conclusion from a different direction: their PAC-Bayes bound for SGD increases with exactly this ratio. This is why we use $r = B/\eta$ as the inverse-randomness parameter for plain SGD, and why the coupled ladder of Appendix G moves B and η together.

The identification is an approximation and not a theorem. The gradient noise is neither Gaussian nor of constant covariance, and away from a minimum the Ornstein–Uhlenbeck picture breaks down. Plain SGD is therefore the algorithm we expect to sit furthest from the Gibbs family, with the largest Δ — the ordering the CIFAR-100 experiment of Section 3 tests.

G Experimental details

G.1 Data

For MNIST we form a binary task by assigning digits 0–4 to one class and digits 5–9 to the other. Each run uses $n = 2000$ training examples. In the random-label condition the labels are drawn independently and uniformly, and the randomization is part of the dataset draw, not of the training run. The dataset draw, including the random labels, is pinned by its own seed and is shared by all optimization seeds, so runs that differ only in the sampler seed see exactly the same data.

For CIFAR-100 we build binary tasks from fine-class pairs drawn without replacement, so no class is reused across tasks. Each task uses $n = 1000$ training and 200 test images.

G.2 Network architectures

The main architecture is a fully connected ReLU network with one hidden layer and a scalar output, so the prediction is the sign of the output. The hidden layer has width 500 for MNIST, where the input is the 784-dimensional flattened image, and width 1000 for CIFAR-100, where the input is 3072-dimensional.

To check that nothing depends on this choice we repeat the MNIST experiment with LeNet-5 [LeCun et al., 1998]. The 28×28 input is padded to 32×32. The network alternates convolution and pooling: C1 applies 6 filters of size 5×5, S2 is a 2×2 average pooling layer, C3 applies 16 filters of size 5×5, S4 is again 2×2 average pooling, and C5 applies 120 filters of size 5×5, which for this input size produces 120 feature maps of size 1×1. Two fully connected layers follow. We widen F6 from the original 84 units to 420 and use a single output unit for the binary task. All activations are ReLU rather than the original tanh.

G.3 Loss functions and prior

The prior π is a centered Gaussian on the weights with standard deviation σ_{prior}.

The main configuration uses the bounded binary cross-entropy (BBCE) of Appendix D of Dziugaite and Roy [2018], with bound $l_{\max} = 4$ and $\sigma_{\text{prior}} = 5.0$. Boundedness matters: Theorem 1 is stated for losses with values in $[0, 1]$, and BBCE is the bounded surrogate that lets the empirical path be read as the $\hat{L}$ of the theory. We plot the unnormalized BBCE; the theory applies to $\hat{L}/l_{\max}$, which rescales β by $l_{\max}$.

We also report ordinary, unbounded BCE, with a smaller $\sigma_{\text{prior}} = 0.1$ so that the training error does not blow up at small β, and weight decay 1.25×10^{-4} for the ULA and SGLD runs.

G.4 Algorithms and parameter grids

All MNIST runs use learning rate 0.01 for ULA and SGLD and three optimization seeds.

For ULA the inverse-temperature grid is $\{0, 128, 256, 500, 1000, 2000, 4000, 8000, 16000\}$. SGLD uses the same grid and batch sizes $B \in \{1, 2, 4, 8, 16, 32, 50\}$. Panel (c) of Figure 2 fixes $B = 16$, panel (d) fixes $\beta = 8000$. For the LeNet-5 version, panels (c) and (d) fix $B = 50$ and $\beta = 16000$.

Plain SGD uses the same batch-size values. Its hot-to-cold path is the coupled ladder $B_k = B_0 c_B^k, \eta_k = \eta_0 c_\eta^k$, with $B_0 = 1$, $\eta_0 = 0.2$, $c_B = 2$ and $c_\eta = 3/4$. The value $B = 50$ is obtained by

the continuous interpolation $\eta(B) = \eta_0 B^{\log(c_\eta)/\log(c_B)}$. The coupled SGD runs use weight decay $\lambda = 2 \times 10^{-3}$ at every rung.

Note that the mini-batch size is a swept parameter here, not a fixed implementation choice: for SGLD it is one of the two ways of moving along the path, and for plain SGD it is the only one.

G.5 Burn-in, stopping rule and reported averages

Each run has a burn-in phase and a sampling phase. We do not fix the length of the burn-in in advance. We watch a smoothed version of the training loss and stop burn-in when it has almost stopped going down. The smoothing is done at every gradient step, and it averages the current mini-batch loss with the previous one:

$$\mathbb{M}_t \;=\; \frac{\alpha}{2}\,\hat{L}(h_t) + \frac{\alpha}{2}\,\hat{L}(h_{t-1}) + (1-\alpha)\,\mathbb{M}_{t-1}, \qquad \mathbb{M}_0 = 1. \tag{18}$$

Once per epoch we look at the last two values and continue while

$$\mathbb{M}_t - \mathbb{M}_{t-1} \;<\; \epsilon,$$

so burn-in ends at the first epoch where this fails. We use $\alpha = 2.5 \times 10^{-4}$ and $\epsilon = -10^{-7}$.

We also require at least 2000 gradient updates before the test is allowed to fire, so that it does not trigger on the flat start of a run. The floor is counted in gradient updates rather than epochs, and it is converted by dividing by the number of batches in an epoch, so every batch size gets the same number of updates. In every run we looked at, the test and not the floor is what ended burn-in.

The sampling phase then runs for as many epochs as burn-in took. The training loss we report for each predictor is a time average of the per-epoch training loss along the recorded trajectory, which is our estimate of $\mathbb{E}_\nu[\hat{L}]$; the test 0–1 error is averaged the same way.

G.6 The CIFAR-100 protocol

For each task and each algorithm, we record the training error of several *hot* predictors and the test error of one *cold* model. For the Langevin algorithms, the hot predictors are $\beta \in \{n/4, n/2, n\}$ against a cold $\beta = 8n$, at full batch for ULA and at $B{=}16$ for SGLD. For plain SGD, the hot predictors are $B \in \{1, 2, 3, 4, 6\}$ on a coupled (batch size, step size) ladder against a cold $B{=}50$.

Across tasks we fit an affine map from the hot predictor to the cold test error and evaluate it leave-one-out, reporting mean absolute error in test-error units. The baseline against which every number must be read is the cold model's *own* training error: if that predicted equally well, heating would buy nothing.

The scores. We fit an affine map from the hot predictor to the cold test error and test it leave-one-out: each task in turn is held out, the map is fitted on the other 49, and we see how far its prediction for the held-out task falls from the truth. We summarise these errors in three ways, because they answer three different questions.

MAE is the average size of the error, in units of 0–1 test error. It answers: how far off is a prediction? It depends on how spread out the tasks are, so we also report $\times$MAE, which is the baseline's MAE divided by the predictor's.

R^2 compares the predictor with a much simpler one: always guessing the average cold test error. $R^2 = 0$ means the two are equally good, and $R^2 = 1$ means the predictor is perfect. Because we score the fit on held-out tasks, R^2 can also be negative, and then it means something definite: the predictor is worse than just guessing the average.

Pearson is the usual correlation coefficient. It is $+1$ when the two quantities lie exactly on a rising straight line, -1 on a falling one, and 0 when there is no straight-line relation between them. It sees only straight lines, and one unusual task can move it a lot.

Spearman is the same coefficient, but computed after replacing every number by its rank among the 50 tasks. So it only asks whether the predictor puts the tasks in the right order. It does not care whether the relation is a straight line, and one unusual task can move its rank by only a few places, so it is much harder to disturb. We report Pearson only in Table 3.

Spearman is the score that matches what we prove. Part (iv) of Theorem 2 says the bound *grows* with $\mathbb{E}_{s\sim\rho}[\psi(s)]$, and we cannot observe $\Delta(r)$ or $\alpha(r)$, so the theory tells us the order of the tasks and not the actual numbers. R^2 and MAE assume in addition that a straight line is the right shape. When the two scores disagree, that gap tells us something: against cold full-batch GD the baseline has $R^2 = -0.11$ but Spearman 0.60, so it does order the tasks somewhat, even though the straight line fitted to it is worse than guessing the average.

How the rows of Table 1 are chosen. The single-arm rows report the minimum-MAE arm of each family, selected on the same 50 tasks; leave-one-out validates the affine map, not the arm choice. The averaged rows are not selected.

H Additional experimental results

This section collects the results that did not fit in Section 3. The settings are those of Appendix G unless stated otherwise.

H.1 Binary MNIST

The four figures below all repeat the experiment of Figure 2, changing one thing at a time: the loss, the way the paths are drawn, the mini-batch size, and the architecture. In every case the same pattern survives, which is why we report it as a property of the algorithms rather than of one setting.

An unbounded loss. Figure 4 repeats Figure 2 with ordinary BCE in place of BBCE. The separation between the two label conditions on the hot part of the path is unchanged, so it does not depend on the bounded surrogate.

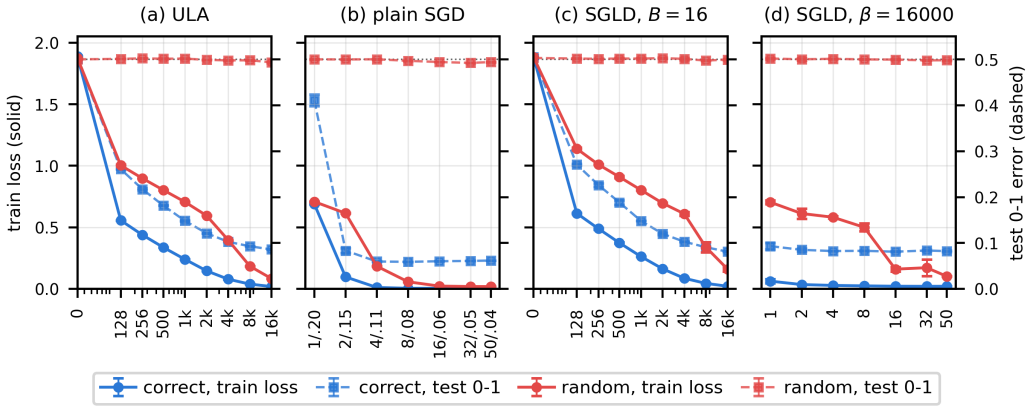


Figure 4: Same as Figure 2, BCE loss ($\sigma_{\text{prior}} = 0.1$) instead of BBCE.

Visualizing label-dependent path separation. Figure 5 is included only to make the difference between the correct-label and random-label empirical paths easier to see. We plot training loss against β on a linear horizontal axis so that the cumulative separation between the two curves is visually apparent. The shaded region therefore represents an empirical comparison of the two label conditions. The curves in this figure are generated by ULA and SGLD, not by exact Gibbs sampling. Consequently, the shaded region must not be interpreted as $\mathrm{KL}(\nu, \pi)$ or as the exact PAC-Bayesian complexity of either algorithm.

SGLD across batch sizes. Figure 2 fixes a single mini-batch size in panel (c), so it does not by itself show whether the pattern is particular to that slice. Figures 6 (BBCE) and 7 (BCE) show the whole SGLD batch-size family on one temperature axis, with full-batch ULA overlaid as the limit the family runs toward. The correct-versus-random separation in hot training loss is present at *every* batch size, and in the correct-label panels the curves are ordered monotonically in B with ULA at the extreme.

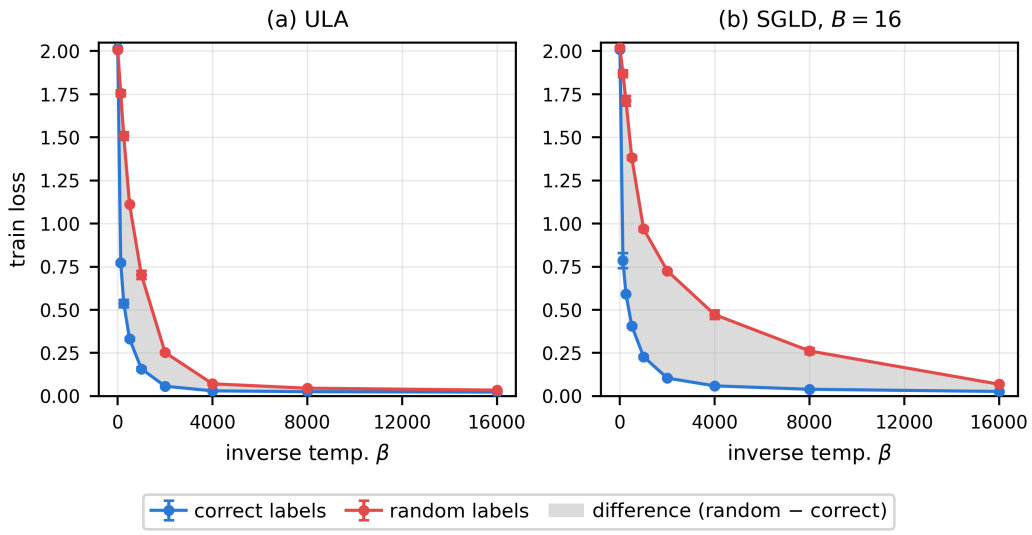


Figure 5: Training-loss paths for correct and random labels under ULA and SGLD with BBCE. The shaded regions visualize the difference between the two empirical paths.

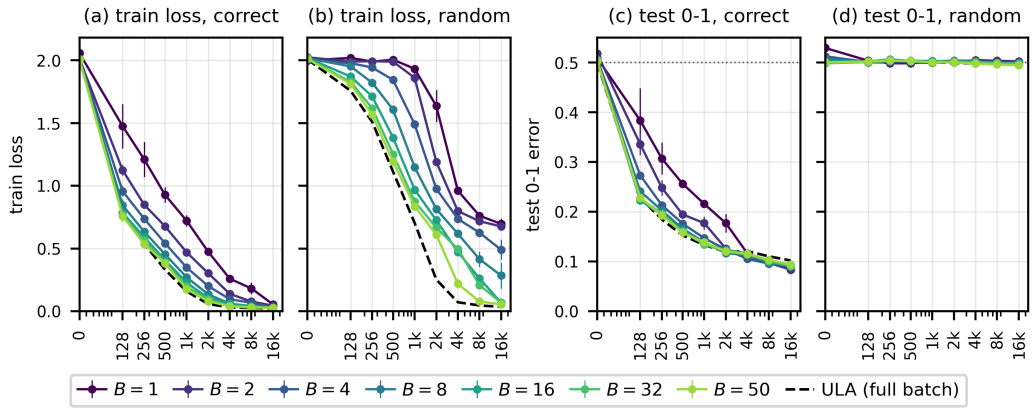


Figure 6: SGLD on binary MNIST, BBCE, mean ± s.e.m. over 3 seeds. Train loss (a),(b) and test 0–1 error (c),(d) against inverse temperature β, one curve per mini-batch size B (sequential colour), with full-batch ULA dashed. Panels (a),(c) are correct labels and (b),(d) random labels; each metric pair shares a y axis. Axis conventions as in Figure 2.

Architecture robustness: LeNet5. Figure 8 repeats the same four panels with LeNet5 in place of the FCN, on the same grids and seeds as Figure 2; panels (c) and (d) fix $B = 50$ and $\beta = 16000$. The pattern is unchanged by the architecture: hot training loss separates the two label conditions well before either test curve moves, and the test curves stay separated at the cold end where the two training losses converge.

H.2 Binary CIFAR-100

Every predictor against every target. Table 3 scores every hot predictor against every cold target, including the cross-algorithm pairings, and reports cost both ways. The gradient-evaluation column counts how many per-sample gradients a run computes; the wall-clock column is the time it actually took. The two can be far apart. Plain SGD at B=3 computes $590\times$ fewer per-sample gradients than full-batch GD, but it only finishes $4.5\times$ faster. A small batch saves arithmetic on each step and then

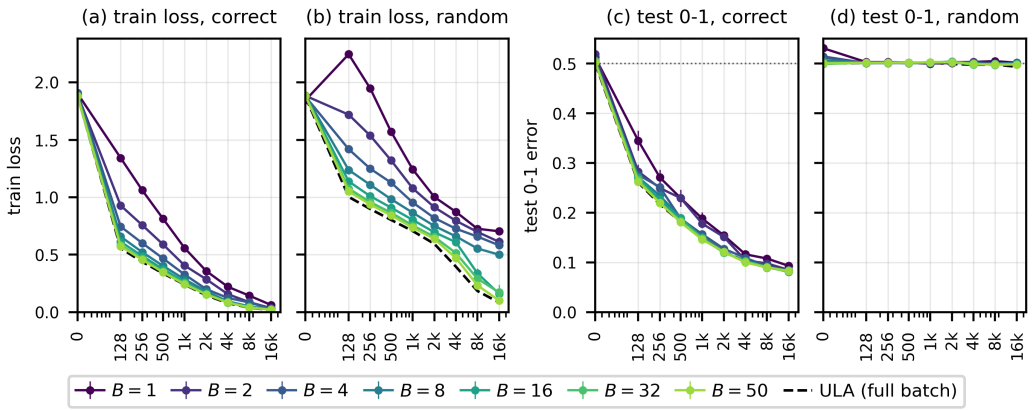


Figure 7: Same as Figure 6, BCE loss ($\sigma_{\text{prior}} = 0.1$) instead of BBCE.

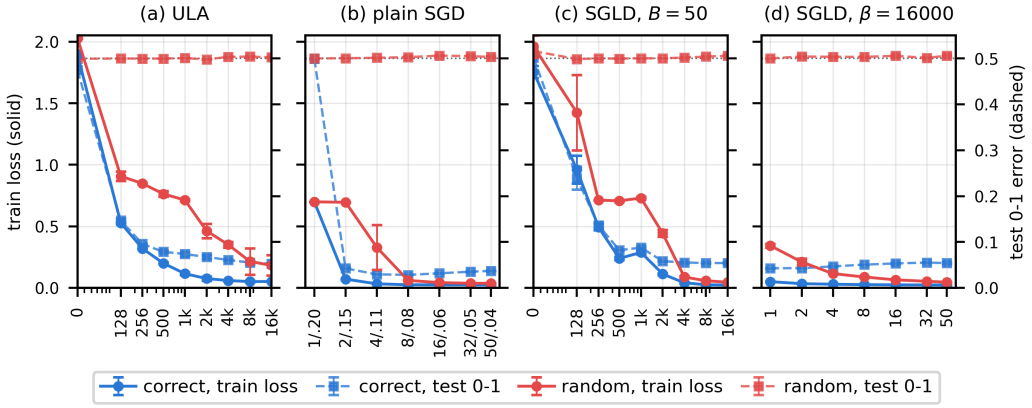


Figure 8: Same as Figure 2 (BBCE, mean $\pm$ s.e.m. over 3 seeds, legend in figure), LeNet5 in place of the FCN. Panel (c) fixes $B = 50$; panel (d) fixes $\beta = 16000$ (vs. $B = 16$, $\beta = 8000$ in Figure 2).

needs many more steps, and those steps have to be taken one after another. Counting gradients hides this, which is why the table in the main text also reports the time.

The single-arm predictors, drawn. Figure 9 draws the four rows of Table 1 as scatter plots, so that the single-arm predictors can be compared by eye with the averaged predictors of Figure 1.

What each predictor costs. A predictor is only useful if it is both accurate and cheap, and the two can be traded against each other. Figure 10 puts every hot predictor we measured on the same pair of axes so that this trade-off is visible in one place.

Each panel is one cold target, and each point is one hot predictor of it. The vertical axis is the leave-one-out MAE, so lower is more accurate. The horizontal axis is how many times cheaper the predictor is than the cold model it predicts, so further right is cheaper, and $x = 1$ is the line where predicting costs as much as simply training the cold model. The horizontal dashed line is the cold model's own training error, the baseline any predictor has to beat. This splits each panel into quadrants: the bottom-right is the useful one, where a predictor is both more accurate than the baseline and cheaper than the target, and the top-left is the useless one. The mini-batch SGLD predictors sit in the bottom right for every target, including the plain-SGD ones.

The cost on the horizontal axis is measured wall-clock, the same quantity as the "wall" column of Table 3, and not a count of gradient evaluations. Counting gradients would flatter the small-batch

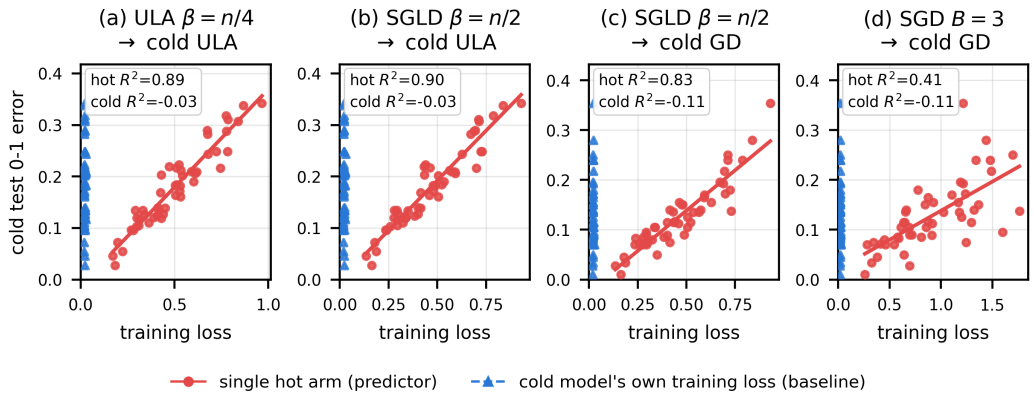


Figure 9: The four rows of Table 1, drawn with the same conventions as Figure 1, so the two can be read side by side. The only difference is that each panel here uses a *single* hot predictor rather than the average over the path: (a) ULA predicting its own cold limit, (b) the same cold ULA predicted instead by one mini-batch SGLD run, (c) that same SGLD run predicting cold full-batch GD, and (d) the plain-SGD ladder predicting its own deterministic limit. One cheap run is close to the averaged predictor of Figure 1 everywhere except plain SGD, whose larger residual Δ is exactly where a single draw should be least sufficient.

predictors, for the reason given above: they compute far fewer gradients but take many more steps one after another, so they do not finish proportionally sooner.

The figure separates the predictor families cleanly. The three Langevin families sit near an MAE of 0.02 in every panel, well under the baseline, and they differ mainly in cost, not in accuracy. The SGD-ladder predictors sit near 0.04, roughly twice the error, in every panel including the one where the cold target is the SGD ladder itself. This is the ordering the residual Δ predicts: the further an algorithm's stationary law is from the Gibbs family, the less its hot predictors say about the cold model.

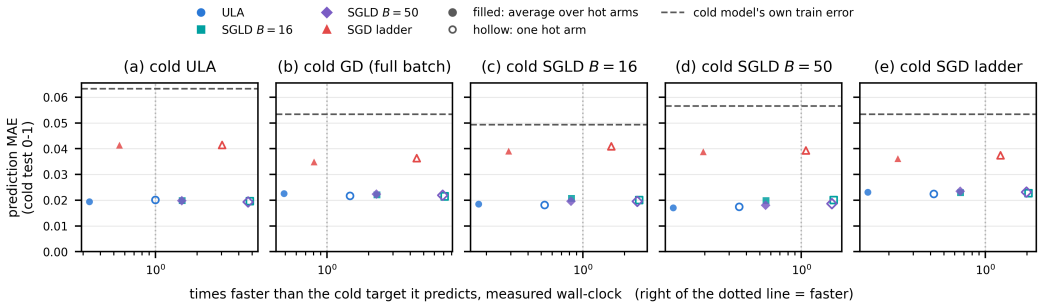


Figure 10: Accuracy against cost for every hot predictor of every cold target, over the 50 binary CIFAR-100 tasks. One panel per cold target; single arms and averaged subsets are both shown. Vertical axis: leave-one-out MAE in predicting cold test 0–1 error, so lower is better. Horizontal axis: how many times faster the predictor is than the cold model it predicts, measured wall-clock, so further right is cheaper. Dashed line: the cold model's own training error, the baseline to beat. Dotted vertical line: equal cost. Every Langevin predictor lies well below the baseline, and the mini-batch SGLD ones also lie to the right of the dotted line in most panels. The SGD-ladder predictors are the only family that stays near the baseline, which is the ordering the residual Δ predicts.

target (cold)	predictor	R^2	Pearson	Spearman	MAE	$\times$MAE	grad.	wall
ULA $\beta{=}8n$	own training error (baseline)	-0.03	0.17	0.30	0.063	1.0	1.0	1.0
	avg. all hot ULA	0.90	0.95	0.95	0.019	3.3	0.5	0.3
	avg. all hot SGLD, $B{=}16$	0.90	0.95	0.94	0.020	3.2	44.8	1.6
	avg. all hot SGLD, $B{=}50$	0.90	0.95	0.95	0.020	3.2	13.3	1.6
	avg. all hot SGD ladder	0.60	0.80	0.79	0.041	1.5	61.4	0.5
	ULA $\beta{=}n/4$	0.89	0.95	0.94	0.020	3.2	1.9	1.0
	SGLD $\beta{=}n/2$, $B{=}16$	0.90	0.95	0.94	0.020	3.2	135.4	4.8
	SGLD $\beta{=}n/2$, $B{=}50$	0.91	0.96	0.95	0.019	3.3	40.6	4.7
	SGD $B{=}3$	0.50	0.74	0.75	0.041	1.5	510.7	3.0
GD (full batch)	own training error (baseline)	-0.11	0.25	0.60	0.053	1.0	1.0	1.0
	avg. all hot ULA	0.82	0.92	0.92	0.023	2.4	0.6	0.5
	avg. all hot SGLD, $B{=}16$	0.82	0.92	0.91	0.022	2.4	51.8	2.3
	avg. all hot SGLD, $B{=}50$	0.82	0.92	0.92	0.022	2.4	15.4	2.3
	avg. all hot SGD ladder	0.51	0.75	0.79	0.035	1.5	71.0	0.8
	ULA $\beta{=}n/4$	0.83	0.92	0.91	0.022	2.5	2.2	1.5
	SGLD $\beta{=}n/2$, $B{=}16$	0.83	0.92	0.91	0.021	2.5	156.6	7.1
	SGLD $\beta{=}n/2$, $B{=}50$	0.82	0.92	0.93	0.022	2.4	46.9	6.9
	SGD $B{=}3$	0.41	0.68	0.74	0.036	1.5	590.5	4.5
SGLD $\beta{=}8n$, $B{=}16$	own training error (baseline)	0.30	0.62	0.67	0.049	1.0	1.0	1.0
	avg. all hot ULA	0.90	0.95	0.95	0.018	2.7	0.0	0.2
	avg. all hot SGLD, $B{=}16$	0.88	0.94	0.93	0.021	2.4	0.9	0.8
	avg. all hot SGLD, $B{=}50$	0.89	0.95	0.94	0.020	2.5	0.3	0.8
	avg. all hot SGD ladder	0.56	0.77	0.80	0.039	1.3	1.2	0.3
	ULA $\beta{=}n/4$	0.90	0.95	0.94	0.018	2.7	0.0	0.5
	SGLD $\beta{=}n/2$, $B{=}16$	0.89	0.95	0.93	0.020	2.5	2.7	2.5
	SGLD $\beta{=}n/2$, $B{=}50$	0.89	0.95	0.94	0.020	2.5	0.8	2.5
	SGD $B{=}3$	0.47	0.72	0.76	0.041	1.2	10.3	1.6
SGLD $\beta{=}8n$, $B{=}50$	own training error (baseline)	0.15	0.47	0.50	0.057	1.0	1.0	1.0
	avg. all hot ULA	0.92	0.96	0.96	0.017	3.3	0.0	0.1
	avg. all hot SGLD, $B{=}16$	0.89	0.95	0.95	0.020	2.9	2.0	0.6
	avg. all hot SGLD, $B{=}50$	0.91	0.96	0.96	0.018	3.1	0.6	0.6
	avg. all hot SGD ladder	0.57	0.78	0.80	0.039	1.5	2.7	0.2
	ULA $\beta{=}n/4$	0.91	0.96	0.95	0.017	3.3	0.1	0.4
	SGLD $\beta{=}n/2$, $B{=}16$	0.89	0.95	0.94	0.020	2.8	5.9	1.7
	SGLD $\beta{=}n/2$, $B{=}50$	0.90	0.96	0.96	0.019	3.0	1.8	1.7
	SGD $B{=}3$	0.50	0.74	0.77	0.039	1.4	22.3	1.1
SGD ladder $B{=}50$	own training error (baseline)	-0.07	0.23	0.59	0.053	1.0	1.0	1.0
	avg. all hot ULA	0.79	0.90	0.91	0.023	2.3	0.0	0.1
	avg. all hot SGLD, $B{=}16$	0.79	0.90	0.89	0.023	2.3	2.3	0.7
	avg. all hot SGLD, $B{=}50$	0.79	0.90	0.90	0.023	2.3	0.7	0.7
	avg. all hot SGD ladder	0.48	0.74	0.78	0.036	1.5	3.2	0.2
	ULA $\beta{=}n/4$	0.80	0.91	0.90	0.022	2.4	0.1	0.4
	SGLD $\beta{=}n/2$, $B{=}16$	0.80	0.91	0.90	0.023	2.4	7.0	2.0
	SGLD $\beta{=}n/2$, $B{=}50$	0.79	0.90	0.91	0.023	2.3	2.1	2.0
	SGD $B{=}3$	0.38	0.66	0.72	0.037	1.4	26.3	1.3

Table 3: Every hot predictor against every cold target, 50 binary CIFAR-100 tasks. R^2, Pearson and Spearman are defined in Appendix G.6; "grad." is the ratio of per-sample gradient evaluations and "wall" the ratio of measured seconds, target to predictor, so below 1 the predictor costs more than the model it predicts.