How Far Can Sharpness and Complexity Jointly Explain Generalization?

Anonymous authors
Paper under double-blind review

Abstract

Large-scale empirical studies have revealed that many widely used generalization measures correlate only weakly with generalization, suggesting that relying on a single factor is insufficient to explain it. In this work, we examine sharpness and complexity jointly, placing the *reach of the two-factor view* at the center: how far can they jointly explain generalization, what determines when they succeed, and where do they fall short? We investigate these questions *quantitatively*, using sixteen alternative metric pairs as probes and Pareto analysis and linear regression as diagnostic tools. Our results show that suitably defined (S, C) pairs provide strong explanatory power across a range of settings. The sharpness perturbation scale substantially affects this success, and the complexity representation influences its robustness. Remaining failures leave open whether more refined measurements are adequate or additional factors are needed.

1 Introduction

Deep networks often achieve strong predictive performance in highly overparameterized regimes, yet why they generalize so well remains incompletely understood (Zhang et al., 2017; Bartlett et al., 2021). A broad body of work has sought more refined explanations of generalization, including the implicit biases induced by architectures and optimization procedures (Neyshabur et al., 2014; Soudry et al., 2018; Gunasekar et al., 2018; Lyu & Li, 2020; Wei & Ma, 2020; Smith et al., 2021), norm- and margin-based bounds (Neyshabur et al., 2015; 2018; Golowich et al., 2020), PAC-Bayes analyses (Dziugaite & Roy, 2017; Neyshabur et al., 2017; Dziugaite et al., 2021), and studies of sharpness and flatness (Hochreiter & Schmidhuber, 1997; Keskar et al., 2017; Chaudhari et al., 2017; Dinh et al., 2017; Tsuzuku et al., 2020; Petzka et al., 2021; Andriushchenko & Flammarion, 2022; Andriushchenko et al., 2023).

Across these diverse approaches, two recurring factors are sharpness and complexity. *Sharpness* $(\boldsymbol{S})$ characterizes the sensitivity of a learned solution to perturbations and is commonly measured using Hessian statistics (Ghorbani et al., 2019; Liu et al., 2022; Kaur et al., 2023) or perturbation-induced increases in loss, as in Sharpness-Aware Minimization (SAM) (Foret et al., 2020; Kwon et al., 2021). *Complexity* $(\boldsymbol{C})$ captures the size or effective capacity of the learned model or its hypothesis class and plays a central role in capacity-based bounds (Bartlett & Mendelson, 2002), PAC and PAC-Bayes theory (McAllester, 1998; 1999; 2003; Guedj, 2019; Alquier, 2024), and many modern generalization analyses (Bartlett et al., 2017; Neyshabur et al., 2018; Arora et al., 2018; Liang et al., 2019; Dziugaite et al., 2020). Common complexity metrics include weight norm, path norm, spectral norm product, and the posterior-prior KL divergence.

Neyshabur et al. (2017) showed that sharpness alone is insufficient to explain several observed generalization phenomena. Motivated by PAC-Bayes theory, they advocated considering sharpness jointly with complexity. Their bi-criteria plots supported this perspective: combining sharpness with ℓ_2 norm complexity accounts for generalization differences that sharpness alone failed to explain.

The subsequent large-scale study by Jiang et al. (2019), however, provided less encouraging evidence for the empirical benefit of combining the two factors. They evaluated 40 generalization measures, including measures that combine sharpness and complexity through a prescribed functional form. Two of these measures, named pacbayes-orig and sharpness-orig in their paper, combine norm complexity with expected sharpness and worst-case sharpness, respectively, to form scalar quantities. They achieved diagnostic scores of 0.360 and 0.398, suggesting limited consistency with the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

observed generalization ordering. Moreover, the authors reported little change in correlation when the norm complexity was removed. This challenges the expected empirical benefit of the second factor in the sharpness-complexity construction advocated by Neyshabur et al. (2017), and casts doubt on whether complexity contributes explanatory information beyond sharpness itself.

These findings leave an important question open: is the two-factor perspective itself inadequate, or is explanatory information lost when sharpness and complexity are combined into a scalar quantity through a prescribed functional form?

To address this, we examine sharpness and complexity jointly *without* committing to a prescribed scalarization. We assess the reach of the two-factor view in a systematic and quantitative manner. Our central question is:

To what extent do sharpness and complexity jointly explain generalization? ($\star$)

The answer to this question turns out to be more nuanced than a simple yes or no, as the extent to which the two factors explain generalization varies across architectures, optimizers, and datasets. Nevertheless, we seek to identify persistent patterns that emerge across most of the settings studied using our framework. To this end, we evaluate sixteen alternative metric pairs across trained models spanning different architectures, training algorithms, and datasets, as well as in mixed settings that pool models from multiple sources. We use Pareto analysis and linear regression as diagnostic tools to characterize the explanatory power of the evaluated metric pairs.

Our **main findings** are summarized below.

i. With suitable metric choices, the two-factor view exhibits strong explanatory power across a broad range of the settings examined, including different architectures, training procedures, datasets, and challenging mixed settings.

ii. Sharpness defined over a larger perturbation scale often yields substantially stronger explanatory power, whereas its classical infinitesimal counterpart can fail. The perturbation scale typically matters much more than the choice of Hessian statistic.

iii. Measuring complexity at the function level yields more robust explanatory power than the parameter-level metrics considered.

iv. Despite being scale-variant, the parameter norm can still be an effective explanatory metric in appropriate training regimes, such as SGD with weight decay.

v. Some regimes remain incompletely explained even by metrics that perform strongly elsewhere.

We *emphasize* that the investigated metric pairs and the diagnostic tools serve as probes for examining the reach of the two-factor view. Other evaluation criteria and alternative metric definitions may also be suitable. Our focus remains on the central question ($\star$).

1.1 Related work

Neyshabur et al. (2017) first connected sharpness and norm-based complexity through a PAC-Bayes framework. Using expected sharpness together with an ℓ_2-norm complexity measure, they showed that incorporating the ℓ_2 norm can improve the explanatory power of sharpness when comparing models trained on true versus random labels across different dataset sizes. Their study was primarily qualitative, aiming to demonstrate the necessity of incorporating a second metric.

In contrast, our work takes a quantitative approach. We systematically evaluate the extent to which (S, C) metric pairs can explain generalization, consider a broad range of alternative definitions for both sharpness and complexity, and study generalization on the original datasets rather than through comparisons with random-label training. Model variation in our experiments is induced by hyperparameter sweeps and different optimizers.

In the large-scale empirical study, Jiang et al. (2019) evaluated more than 40 generalization measures across over 10,000 trained models. Among these, the pacbayes-orig and sharpness-orig incorporate both factors. They combine expected sharpness and worst-case sharpness with norm complexity, respectively, through prescribed functional forms. The authors assessed the correlation between each resulting scalar quantity and generalization, obtaining granulated Kendall scores of 0.360 and 0.398, respectively. These scores suggest that the resulting scalar measures have limited explanatory power, which challenges the expected empirical benefits of the joint perspective. In contrast, we study the pair (S, C) directly, *without prescribing a functional form* for combining the two factors.

Under this two-dimensional view, our empirical findings are substantially more positive, suggesting that part of the limitation observed in prior studies may arise from the scalarization itself rather than from the two-factor perspective.

1.2 NOTATION

Let $Z = \{z_i\}_{i=1}^m$ be a training dataset of m independent samples $z_i = (\mathbf{x}_i, y_i)$ from an unknown distribution $\mathcal{D}$, where $\mathbf{x}_i$ and y_i denote the input and label. For a model h and loss function $\ell(h; z)$, the empirical (or training) and population losses are $\ell(h; Z) := \frac{1}{m}\sum_{i=1}^m \ell(h; z_i)$ and $\ell(h; \mathcal{D}) := \mathbb{E}_{z\sim\mathcal{D}}[\ell(h; z)]$. The generalization gap measures the increase from training to population loss, $\mathrm{Gap}(h) := \ell(h; \mathcal{D}) - \ell(h; Z)$. Since $\mathcal{D}$ is unknown, this gap is commonly estimated by $\widehat{\mathrm{Gap}}(h) := \text{test loss} - \text{training loss}$. We use w to denote the network parameter vector and f_w its corresponding function.

2 A KEY DISTINCTION: EXPLANATION VS. REGULARIZATION

Many quantities related to generalization first attract practical interest because they work well as training regularizers. Table 1 lists several widely used examples.

Table 1: Common regularizers.

Regularizer	Generalization-related factor
noise injection	sharpness
SAM	sharpness
weight decay	complexity

Weight decay in SGD provides a familiar example: regularizing the squared weight norm, $\|w\|_2^2$, often improves test performance (Krogh & Hertz, 1991; Loshchilov & Hutter, 2019; Zhang et al., 2019; D'Angelo et al., 2024). However, $\|w\|_2^2$ itself is not always a reliable complexity metric. For networks that admit function-preserving rescalings, the weights can be rescaled so that $\|w\|_2^2$ changes substantially while the network function and its generalization behavior remain unchanged. This highlights an important ***distinction between successful regularization, well-posedness as a metric, and sufficient explanatory power***. A quantity's effectiveness as a regularizer and its reliability as an explanatory metric are related but distinct properties, and therefore motivate different research questions.

From the perspective of regularization, the question is: *does penalizing $\|w\|_2^2$ in training help yield a better-generalizing model in a particular setting?* The explanatory analysis in this work asks: *is a smaller $\|w\|_2^2$ consistently associated with better generalization across many trained models?*

These two roles are clearly related: if a quantity exhibits strong explanatory power across many trained models, we would generally expect it to be a promising regularizer across a range of individual settings, although it need not be the best regularizer in every particular case. The converse, however, does not necessarily hold: a quantity may serve as an effective regularizer in certain settings without providing a reliable explanatory factor across a broader population of trained models.

3 (S, C) METRIC PAIRS

3.1 SCALING INVARIANCE

Deep networks exhibit *parameterization ambiguity*: distinct parameter vectors w and w' can represent exactly the same function, $f_{w'} = f_w$. A common example is *scaling ambiguity*, which arises from compensating layer-wise rescalings permitted by positively homogeneous activations. For instance, in a ReLU network, multiplying the weights of one layer by $\alpha > 0$ and dividing the weights of the subsequent layer by α leaves the represented function unchanged, while potentially changing parameter-dependent quantities substantially.

Ideally, a generalization metric should be invariant to such function-preserving rescalings, since the network function, and hence its predictive performance, remains unchanged. Nevertheless, we do not restrict our study exclusively to scale-invariant metrics. In highly overparameterized neural networks, rescaling represents only one source of parameterization ambiguity. Consequently, scale invariance alone does not guarantee invariance to more general function-preserving reparameterizations, nor does it by itself ensure that a metric is well posed. We therefore regard ***scale invariance*** as a desirable property rather than a necessary criterion for including a metric in our study.

3.2 SIXTEEN METRIC PAIRS – PROBES FOR STUDYING THE EXPLANATORY POWER

Four Sharpness Metrics. Sharpness describes the geometry of the loss landscape around a learned solution: a solution is considered sharp if small parameter perturbations can cause large increases in loss. Local curvature is commonly quantified by Hessian statistics, such as the trace and the top eigenvalue. However, the ordinary Hessian depends on the parameterization. As noted above, certain parameter rescalings can change these statistics without affecting the model's generalization performance (Dinh et al., 2017). To reduce scaling ambiguity, we consider sharpness metrics that are *adaptive to parameter magnitudes*. That is to say, throughout this paper, unless specified otherwise, we compute sharpness metrics using the adaptive Hessian:

$$\mathrm{H} = \mathrm{diag}(|w|)\, \nabla_w^2 \ell(f_w; Z)\, \mathrm{diag}(|w|).$$

We consider both classical infinitesimal sharpness metrics and their finite-scale counterparts, motivated by our observation that their explanatory power can differ substantially. The infinitesimal metrics are estimated directly from the adaptive Hessian, while the finite-scale metrics are evaluated through perturbed losses.

- The Hessian trace, $\mathrm{trace}(\mathrm{H})$, *summarizes* loss curvature across parameter directions, capturing second-order changes in the infinitesimal perturbation limit. We estimate it using the method of Hutchinson (1989).
- The Hessian top eigenvalue, $\lambda_{\max}(\mathrm{H})$, measures the *largest* infinitesimal loss curvature among parameter directions. We estimate it using Lanczos (1950) iteration.
- The finite-scale Hessian trace,

$$\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) := \mathbb{E}_\epsilon \frac{[L(w+\sigma\epsilon) - L(w)]}{\sigma^2}, \quad \epsilon = |w| \odot z, \quad z \sim \mathcal{N}(0, I). \tag{1}$$

  A second-order Taylor expansion gives $\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) \to \frac{1}{2}\,\mathrm{tr}(H)$ as $\sigma \to 0$. Thus, up to a constant factor, this finite-scale $\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$ approaches the infinitesimal $\mathrm{trace}(\mathrm{H})$ as the perturbation scale approaches zero (Appendix A.3). We *estimate the expectation* by Monte Carlo sampling: draw independent anisotropic Gaussian perturbations, average loss increases, and normalize by σ^2.
- The finite-scale Hessian top eigenvalue,

$$\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) := \max_{\|v\|_2=1} \frac{L(w+\sigma|w|\odot v) + L(w-\sigma|w|\odot v) - 2L(w)}{\sigma^2}. \tag{2}$$

  A second-order Taylor expansion shows that symmetry cancels the linear terms, leaving the limiting quadratic form $v^\top H v$ after normalization. Maximizing over unit directions yields the adaptive Hessian top eigenvalue: $\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) \to \lambda_{\max}(H)$ as $\sigma \to 0$ (Appendix A.3). We *approximate the maximum* using a number of probe directions, refined through gradient ascent, and retain the largest value found.

Four Complexity Metrics.

- Parameter Norm. The classical norm-based complexity metric, $\|w\|_2$, corresponds to the common regularizer, weight decay in SGD. Parameter norm is scale-*variant* by definition.
- Function Norm. As discussed in Section 3.1, scaling ambiguity is just one type of parameterization ambiguity. To reduce the parameterization ambiguity as a whole, we define a function-level complexity metric. Consider a network parameterized by w, and write f_w for its represented function. For classification tasks, given a fixed reference dataset $D_{\mathrm{ref}} = \{\mathbf{x}_i\}_{i=1}^n$, let Z_w denote the logit matrix:

$$Z_w(D_{\mathrm{ref}}) := \begin{bmatrix} f_w(\mathbf{x}_1)^\top \\ \vdots \\ f_w(\mathbf{x}_n)^\top \end{bmatrix} \in \mathbb{R}^{n\times c},$$

  where $f_w(\mathbf{x}_i)$ is the c-class logit vector for $\mathbf{x}_i$. Softmax probabilities are invariant under the transformation $Z_w \mapsto Z_w + \mathbf{e}\mathbf{1}_c^\top$ for any $\mathbf{e} \in \mathbb{R}^n$. We remove this additive ambiguity using the class-centering projection $\Pi_c = I_c - \frac{1}{c}\mathbf{1}_c\mathbf{1}_c^\top$. The *class-centered logit matrix* is then defined as

$$\widetilde{Z}_w(D_{\mathrm{ref}}) := Z_w(D_{\mathrm{ref}})\Pi_c = \begin{bmatrix} \widetilde{f}_w(\mathbf{x}_1)^\top \\ \vdots \\ \widetilde{f}_w(\mathbf{x}_n)^\top \end{bmatrix} \in \mathbb{R}^{n\times c},$$

where $\widetilde{f}_w(\mathbf{x}_i) = f_w(\mathbf{x}_i) - \frac{1}{c}\mathbf{1}_c\mathbf{1}_c^\top f_w(\mathbf{x}_i)$. We define the *function norm* as

$$\text{func norm} := \left\|\widetilde{Z}_w(D_{\text{ref}})\right\|_F. \tag{3}$$

This quantity is not only scale-invariant but also parameterization-*invariant*. It can be interpreted as a *proxy* for a simplified posterior-prior KL divergence in function space; see Appendix A.4.

- Path Norm (Neyshabur et al., 2015). Let $\mathcal{V}$ denote all input–output paths and $\pi_v(w) = \prod_{i\in v} w_i$ the product of multiplicative parameters along path v. The path norm is

$$\text{path norm} := \Big(\sum_{v\in\mathcal{V}} \pi_v(w)^2\Big)^{1/2} = \Big(\sum_{v\in\mathcal{V}} \prod_{i\in v} w_i^2\Big)^{1/2}. \tag{4}$$

  Unlike parameter norm, path norm is scale-*invariant*. In practice, we report the depth-normalized quantity $\text{path norm}^{1/L_{\text{eff}}}$, where L_{eff} counts effective multiplicative layers. This normalization reduces depth-dependence and facilitates comparisons across networks.

- Spectral Norm Product (Bartlett et al., 2017). For an L-layer network with weight matrices $W_1, \ldots, W_L$, the spectral norm product is

$$\text{spec norm} := \prod_{j=1}^{L} \|W_j\|_2, \tag{5}$$

  where $\|W_j\|_2$ is the spectral norm of the weight matrix at layer j. This quantity is scale-*invariant*, with smaller values suggesting a more controlled input-output mapping. We likewise report the depth-normalized quantity $\text{spec norm}^{1/L_{\text{eff}}}$, with L_{eff} defined above.

Table 2 summarizes the evaluated metrics and their properties. We evaluate $4 \times 4 = 16$ metric pairs obtained by combining the four sharpness metrics with the four complexity metrics. **Each metric pair *serves as a probe* to investigate our goal – the reach of the two-factor view.**

Table 2: Evaluated metrics and their properties.

Sharpness metrics		**Complexity metrics**	
trace(H)	scale-invariant	param norm	scale-*variant*
$\lambda_{\max}\text{(H)}$	scale-invariant	func norm	parameterization-invariant
$\text{trace(H)}_{\text{fs}}$	scale-invariant	path norm	scale-invariant
$\lambda_{\max}\text{(H)}_{\text{fs}}$	scale-invariant	spec norm	scale-invariant

4 Evaluation tools

Linear Regression. For a fixed architecture and training procedure, we sweep hyperparameters to obtain *a collection of well-trained models*, each serving as one regression point. For model i, let S_i and C_i denote its sharpness and complexity under the evaluated metric pair, and Gap_i its generalization gap (test loss minus training loss). We fit

$$\text{Gap}_i = \beta_0 + \beta_S S_i + \beta_C C_i + \varepsilon_i, \qquad i = 1, \ldots, N,$$

where N is the collection size and ε_i is the regression residual. We report R^2 and the signs of β_S and β_C in Section 5. A high R^2 with positive coefficients indicates strong explanatory power under a linear approximation.

Linear regression provides a straightforward diagnostic but is limited to linear relationships. Our goal, however, is to assess the joint view *without* prescribing a functional form (Section 1.1). A poor linear fit may reflect nonlinearity rather than failure of the two-factor view. This motivates our **methodological contribution**: an order-based diagnostic tool that directly tests the monotonic ordering implied by the two-factor view without prescribing a functional form.

Pareto Analysis. To avoid imposing any functional combination, we introduce Pareto analysis. The main idea is to compare the ordering induced by (S, C) with the ordering of generalization gaps. This analysis relies only on the ***monotonic relationship*** suggested by the two-factor view:

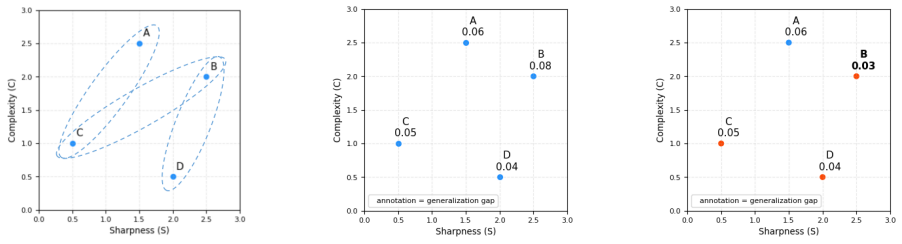
if model i has both lower S and lower C than model j, then model i should generalize better.

Given two trained models i and j, we call (i, j) a **candidate pair** if model i has both lower S and lower C than model j, i.e., $S_i < S_j$ and $C_i < C_j$. Based on the expected monotonic relationship, this ordering implies $\text{Gap}_i < \text{Gap}_j$. We focus on such candidate pairs because only these pairs induce a

clear expected ordering from the monotonicity. If one model has lower S but higher C than the other, no clear generalization ordering follows without specifying a particular functional form, which we do not impose in this paper. A candidate pair is called a **contradictory pair** if this expected ordering is violated: $S_i < S_j, C_i < C_j$, but $\mathrm{Gap}_i > \mathrm{Gap}_j$. Such a contradictory pair indicates that the ordering induced by the evaluated metric pair *disagrees* with the generalization ordering. The frequency of such violations is quantified by ***Pareto contradiction rate***,

$$\mathrm{PCR} := \frac{\#\{\text{contradictory pairs}\}}{\#\{\text{candidate pairs}\}}.$$

A high PCR means frequent violations of the monotonic relationship expected from the two-factor view. Conversely, *a lower PCR indicates better agreement* between the (S, C)-induced ordering and the generalization ordering, and therefore *stronger* explanatory power of the metric pair evaluated. Figure 1 illustrates how PCR is computed. Numerical PCR results are reported in Section 5. Visualized PCRs are presented in Figure 4 and Figure 5 in the Appendix.



(a) *candidate pairs*: {AC,BC,BD}, count 3.

(b) *contradictory pairs*: None, count 0. PCR $= \frac{0}{3} = 0\%$.

(c) *contradictory pairs*:{BC,BD}, count 2. PCR $= \frac{2}{3} = 66.7\%$.

Figure 1: Computing PCR for an artificial collection of trained models $\{A, B, C, D\}$. Each point denotes a trained model, annotated by its generalization gap. The middle: no violations occur, yielding low PCR of 0%. The right: frequent violations occur, yielding high PCR of 66.7%.

Diagnostic Criteria: interpreting two evaluation tools together. Pareto analysis and linear regression provide complementary diagnostics of the explanatory power of an evaluated metric pair. PCR measures violations of the monotonic ordering predicted by the two-factor view (*lower is better*), while R^2 evaluates how well a linear model captures the relationship (*higher is better*). The outcomes of interpreting their diagnostics jointly are summarized in Table 3.

Table 3: Joint interpretation of Pareto analysis and linear regression diagnostics.

Diagnostics of Two-Factor Explanatory Power		**Interpretation**
PCR ↓	**Linear regression R^2 ↑, coeffi. signs**	
low	high; positive	order-consistent, approximately linear
low	low; positive	order-consistent, likely nonlinear
high	high; positive	order-inconsistent, despite linear trend
high	low; positive	order-inconsistent, no linear trend
any	any; at least one negative	directionally inconsistent with two-factor view

We ***primarily use*** PCR to assess explanatory power because it does not assume a functional relationship between S, C, and the generalization gap. We *retain* linear regression as it provides a complementary diagnostic of whether the relationship is approximately additive. We regard low PCR as favorable, and high R^2 together with positive regression coefficients provides additional evidence of an approximately linear relationship. See Appendix B for a detailed discussion.

5 Numerical Results and Observations

Experimental Setup. We evaluate the explanatory power of the two-factor view across a diverse collection of trained networks. For each experimental setting, we fix the dataset, architecture, and training algorithm, and obtain a population of trained models by varying hyperparameters. We then retain those well-trained models, selected by sufficiently small training loss; see Appendix C for details. For every well-trained model, we record its generalization gap (test loss minus training loss) and compute its sharpness and complexity values under the evaluated metrics described in Section 3. We then assess how well each metric pair explains the variation in generalization gaps across the resulting model population using two complementary diagnostics: (i) PCR, which measures how frequently the expected ordering of generalization gaps is violated; and (ii) R^2 together with the

coefficient signs from linear regression of the generalization gap. We repeat this analysis across different architectures, datasets, and training procedures. We additionally consider mixed settings, in which models from multiple sources are pooled together, to evaluate the robustness of the observed patterns. The main results are reported in Table 4, 5, 7, Figure 2, and Figure 3.

Table 4: **Main results.** Diagnostics of two-factor explanatory power by Pareto analysis and linear regression. Each vertical block, separated by dashed lines, corresponds to a fixed architecture, optimizer, and dataset. For each block, models are generated by varying hyperparameters, and sixteen metric pairs are evaluated using PCR based on generalization-gap rankings and R^2 from linear regression against generalization gap. *Low* PCR is prioritized as evidence of *strong explanatory power*, and high R^2 with positive coefficients further supports an approximately linear relationship (Section 4). Finite-scale sharpness metrics use the large perturbation scale $\sigma = 0.50$. Path norm and spectral norm lack natural extensions to ViT, so the corresponding entries are left blank. See visualized PCRs in Figure 5.

(S, C) Metric Pair	**ResNet18** sgd		**VGG13** sgd		**ResNet18 no-BN** sgd		**VGG13 no-BN** sgd	
	CIFAR-10							
	PCR	$\boldsymbol{R^2}\,(S,C)$	**PCR** PCR↓: lower is better	$\boldsymbol{R^2}\,(S,C)$	**PCR**	$\boldsymbol{R^2}\,(S,C)$ R^2↑: higher with $(+,+)$ is better	**PCR**	$\boldsymbol{R^2}\,(S,C)$
$(\mathrm{trace(H)}$, func norm$)$	1.2%	$0.48\,(+,+)$	8.2%	$0.31\,(+,+)$	4.1%	$0.79\,(+,+)$	5.4%	$0.68\,(+,+)$
$(\lambda_{\max}\mathrm{(H)}$, func norm$)$	2.6%	$0.45\,(+,+)$	8.4%	$0.31\,(+,+)$	3.9%	$0.79\,(+,+)$	5.8%	$0.68\,(+,+)$
$(trace(H)_{fs}$, func norm$)$	1.5%	$0.95\,(+,+)$	2.8%	$0.97\,(+,+)$	1.5%	$0.85\,(+,+)$	3.4%	$0.85\,(+,+)$
$(\lambda_{max}(H)_{fs}$, func norm$)$	1.9%	$0.92\,(+,+)$	4.5%	$0.87\,(+,+)$	1.7%	$0.82\,(+,+)$	4.3%	$0.85\,(+,+)$
$(\mathrm{trace(H)}$, param norm$)$	3.5%	$0.84\,(+,+)$	13.4%	$0.70\,(+,+)$	49.5%	$0.21\,(-,+)$	61.0%	$0.42\,(-,+)$
$(\lambda_{\max}\mathrm{(H)}$, param norm$)$	4.4%	$0.84\,(+,+)$	13.0%	$0.69\,(+,+)$	33.0%	$0.20\,(-,+)$	59.0%	$0.42\,(+,+)$
$(trace(H)_{fs}$, param norm$)$	3.0%	$0.98\,(+,+)$	6.0%	$0.95\,(+,+)$	3.4%	$0.68\,(+,+)$	3.0%	$0.72\,(+,-)$
$(\lambda_{max}(H)_{fs}$, param norm$)$	3.6%	$0.97\,(+,+)$	8.5%	$0.92\,(+,+)$	6.0%	$0.56\,(+,+)$	5.6%	$0.70\,(+,-)$
$(\mathrm{trace(H)}$, path norm$)$	15.1%	$0.90\,(+,+)$	31.1%	$0.85\,(-,+)$	61.0%	$0.19\,(-,+)$	66.7%	$0.42\,(-,+)$
$(\lambda_{\max}\mathrm{(H)}$, path norm$)$	15.7%	$0.90\,(+,+)$	31.0%	$0.84\,(-,+)$	47.2%	$0.18\,(-,+)$	68.3%	$0.42\,(-,+)$
$(trace(H)_{fs}$, path norm$)$	8.6%	$0.96\,(+,+)$	16.8%	$0.86\,(+,+)$	5.1%	$0.68\,(+,+)$	3.2%	$0.72\,(+,-)$
$(\lambda_{max}(H)_{fs}$, path norm$)$	9.9%	$0.96\,(+,+)$	19.6%	$0.84\,(+,+)$	7.7%	$0.56\,(+,+)$	6.1%	$0.70\,(+,-)$
$(\mathrm{trace(H)}$, spec norm$)$	54.0%	$0.07\,(+,+)$	53.6%	$0.17\,(-,+)$	80.6%	$0.20\,(-,+)$	69.1%	$0.44\,(-,+)$
$(\lambda_{\max}\mathrm{(H)}$, spec norm$)$	55.3%	$0.05\,(+,+)$	54.1%	$0.17\,(-,+)$	74.0%	$0.17\,(-,+)$	72.2%	$0.44\,(-,+)$
$(trace(H)_{fs}$, spec norm$)$	14.0%	$0.88\,(+,-)$	21.0%	$0.83\,(+,-)$	4.4%	$0.66\,(+,+)$	3.4%	$0.75\,(+,-)$
$(\lambda_{max}(H)_{fs}$, spec norm$)$	30.5%	$0.82\,(+,-)$	31.4%	$0.58\,(+,-)$	7.8%	$0.53\,(+,+)$	6.7%	$0.73\,(+,-)$
(S, C) Metric Pair	**WideResNet** sgd		**ViT** adamw		**WideResNet no-BN** sgd		**ViT no-LN** adamw	
	CIFAR-100							
	PCR	$\boldsymbol{R^2}\,(S,C)$	**PCR**	$\boldsymbol{R^2}\,(S,C)$	**PCR**	$\boldsymbol{R^2}\,(S,C)$	**PCR**	$\boldsymbol{R^2}\,(S,C)$
$(\mathrm{trace(H)}$, func norm$)$	4.6%	$0.88\,(+,+)$	40.6%	$0.80\,(-,+)$	27.7%	$0.40\,(+,+)$	16.7%	$0.70\,(+,+)$
$(\lambda_{\max}\mathrm{(H)}$, func norm$)$	2.9%	$0.88\,(+,+)$	26.2%	$0.80\,(-,+)$	31.1%	$0.29\,(+,+)$	16.5%	$0.72\,(+,+)$
$(trace(H)_{fs}$, func norm$)$	1.7%	$0.95\,(+,+)$	9.4%	$0.82\,(+,+)$	31.3%	$0.32\,(-,+)$	12.8%	$0.74\,(+,+)$
$(\lambda_{max}(H)_{fs}$, func norm$)$	1.8%	$0.92\,(+,+)$	14.5%	$0.80\,(-,+)$	31.5%	$0.36\,(-,+)$	13.2%	$0.74\,(+,+)$
$(\mathrm{trace(H)}$, param norm$)$	2.4%	$0.93\,(+,+)$	39.8%	$0.78\,(-,+)$	51.0%	$0.05\,(+,+)$	24.5%	$0.04\,(+,+)$
$(\lambda_{\max}\mathrm{(H)}$, param norm$)$	1.5%	$0.91\,(+,+)$	18.3%	$0.78\,(-,+)$	54.4%	$0.01\,(-,-)$	17.9%	$0.13\,(+,+)$
$(trace(H)_{fs}$, param norm$)$	2.1%	$0.96\,(+,+)$	7.0%	$0.78\,(+,+)$	40.0%	$0.01\,(-,+)$	39.5%	$0.47\,(+,-)$
$(\lambda_{max}(H)_{fs}$, param norm$)$	2.3%	$0.95\,(+,+)$	11.0%	$0.78\,(-,+)$	41.4%	$0.04\,(-,+)$	20.2%	$0.35\,(+,-)$
$(\mathrm{trace(H)}$, path norm$)$	12.8%	$0.90\,(+,+)$	–	–	56.9%	$0.03\,(+,+)$	–	–
$(\lambda_{\max}\mathrm{(H)}$, path norm$)$	8.4%	$0.91\,(+,+)$	–	–	61.8%	$0.02\,(-,-)$	–	–
$(trace(H)_{fs}$, path norm$)$	4.7%	$0.96\,(+,+)$	–	–	41.9%	$0.01\,(+,-)$	–	–
$(\lambda_{max}(H)_{fs}$, path norm$)$	4.3%	$0.95\,(+,+)$	–	–	43.3%	$0.03\,(-,-)$	–	–
$(\mathrm{trace(H)}$, spec norm$)$	47.4%	$0.43\,(+,-)$	–	–	53.0%	$0.03\,(+,+)$	–	–
$(\lambda_{\max}\mathrm{(H)}$, spec norm$)$	43.9%	$0.53\,(+,-)$	–	–	57.4%	$0.03\,(-,-)$	–	–
$(trace(H)_{fs}$, spec norm$)$	5.8%	$0.90\,(+,-)$	–	–	38.5%	$0.01\,(-,-)$	–	–
$(\lambda_{max}(H)_{fs}$, spec norm$)$	10.8%	$0.83\,(+,-)$	–	–	39.6%	$0.05\,(-,-)$	–	–

Interesting observations based on the results.

1. **The two-factor view provides *strong* explanatory power across different settings.** Overall, our findings are encouraging: with appropriate metric choices, sharpness and complexity can jointly account for a substantial fraction of the variation in generalization across a range of settings. The results in Table 4, 5, 7, Figure 2 and 3 make this answer concrete. For nearly every setting evaluated, at least one metric pair exhibits strong explanatory power. WideResNet no-BN is the only exception where no evaluated metric pair shows strong explanatory power (Observation 8).
2. **Sharpness defined over a *larger* neighborhood has much stronger explanatory power.** Conventional sharpness metrics, $\mathrm{trace(H)}$ and $\lambda_{\max}\mathrm{(H)}$, characterize local loss changes as the per-

turbation radius approaches zero and show relatively weak explanatory power. In contrast, their finite-scale counterparts, $\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$ and $\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$, can provide substantially stronger explanatory power at larger perturbation scales. Holding the complexity metric fixed clearly reveals this *infinitesimal versus finite-scale* contrast. For ResNet18 no-BN in Table 4, with parameter norm as complexity, replacing $\mathrm{trace}(\mathrm{H})$ with $\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$ *reduces* PCR from 49.5% to 3.4% and *increases* R^2 from 0.21 to 0.68. We ***further examine*** how PCR changes as the sharpness perturbation radius increases from an *infinitesimal* scale to *small* and then *large* finite scale. Figure 2 and 3 show this progression, with the complexity metric fixed in each subplot. For sharpness based on either Hessian trace (blue circle) or top eigenvalue (pink triangle), PCR often decreases at larger scales, indicating stronger explanatory power. This trend is especially pronounced when sharpness metric is paired with parameter norm, path norm, and spectral norm.

3. **The *perturbation scale* matters much more than the choice of Hessian statistic.** In most four-by-two mini-blocks in the tables, which compare four sharpness metrics at a fixed complexity metric, the trace and top eigenvalue often give similar diagnostics, while *the improvement from infinitesimal to finite-scale sharpness is much larger*. This suggests that the scale at which the loss landscape is probed plays a substantially greater role than the way its curvature is summarized. For example, in the parameter norm mini-block for sgd-trained VGG13 no-BN in Table 4, $\mathrm{trace}(\mathrm{H})$ and $\lambda_{\max}(\mathrm{H})$ produce PCRs 61.0% and 59.0%, while their finite-scale counterparts, $\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$ and $\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$, yield much lower PCRs 3.0% and 5.6%. Figure 2 and 3 show more visible evidence for this observation. PCR differences between $\mathrm{trace}(\mathrm{H})$ (blue circle) and $\lambda_{\max}(\mathrm{H})$ (pink triangle) are often small or even tiny, whereas those between small-scale sharpness $(\sigma \leq 0.10)$ and large-scale sharpness $(\sigma \geq 0.50)$ can be substantial.

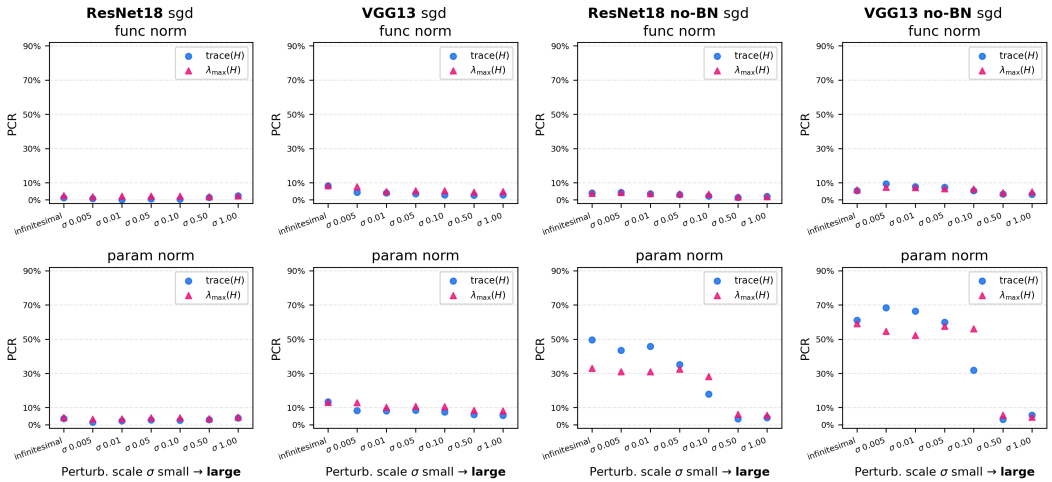


Figure 2: How PCR varies as the sharpness perturbation radius increases from an infinitesimal scale to small and then large finite scale. Each column corresponds to an architecture. In each subplot, with the complexity metric fixed, sharpness defined over a *larger* neighborhood yields lower PCR, indicating stronger explanatory power and supporting Observation 2. The row-wise comparisons support Observations 4, 5, and 6. Results of path norm and spectral norm are provided in Figure 3.

4. **Function norm exhibits *more stable* explanatory power than parameter norm.** This stability appears in two ways. **First**, the explanatory power of function norm is *less sensitive to sharpness choice*, particularly infinitesimal versus finite-scale sharpness. For VGG13 no-BN in Table 4, function norm maintains strong explanatory power with both $\lambda_{\max}(\mathrm{H})$ and $\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$ (PCRs 5.8% and 4.3%, respectively), whereas parameter norm varies markedly (59.0% and 5.6%). The function norm subplots in the first row of Figure 2 likewise show highly consistent explanatory power across sharpness choices and scales, from infinitesimal to large. **Second**, function norm retains strong and robust explanatory power when pooling models across architectures, optimizers, or datasets into *mixed settings*, whereas parameter norm can collapse; Table 5 shows this contrast. Further mixed setting observations are discussed in Observation 7 below.
5. **Although scale-*variant*, parameter norm can still explain generalization effectively in appropriate training regimes.** SGD with weight decay may encourage balanced weight scales during training, limiting the influence of parameterization ambiguity. In such regimes, lacking scale invariance does not necessarily weaken the explanatory power of parameter norm. For example,

sgd-trained WideResNet in Table 4, pairing $\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$ with parameter norm gives R^2 of 0.95, versus 0.92 with function norm, and PCRs are below 3% for both.

6. **Path norm and spectral norm exhibit less consistent explanatory power across settings.** Although both can perform well in certain individual settings, their diagnostics vary considerably across architectures and training settings. Many corresponding pairs exhibit large PCR, lower R^2, or negative coefficient signs, especially with infinitesimal sharpness, as seen for sgd-trained ResNet18 no-BN in Table 4. Finite-scale sharpness often improves these diagnostics, but they remain less consistent across settings than those obtained with function norm.
7. **The two-factor explanatory power *remains strong* when pooling across architectures, optimizers and even datasets, but this depends strongly on the choice of complexity metric.** Some metric pairs perform well within individual settings (Tables 4 and 7) but deteriorate substantially in mixed collections (Table 5). Pairs involving function norm remain much more stable: paired with finite-scale sharpness, function norm consistently yields low PCR, whereas parameter norm can yield much higher PCR in the mixed settings.
8. **There still exist regimes that even the strongest metric pairs do not fully explain.** For WideResNet no-BN in Table 4, finite-scale sharpness paired with function norm still gives weak diagnostics: $(\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}, \text{func norm})$ yields PCR 31.3% and R^2 0.32, with similarly weak results for $(\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}, \text{func norm})$. This suggests that, despite strong explanatory power across many settings, the two-factor view with the evaluated metrics does not yet fully explain generalization.

Table 5: **Main results** (mixed settings). Metric pairs and diagnostic criteria are the same as in Table 4; here each vertical block reports diagnostics for *a mixed setting*: pooling models trained from multiple sources into one collection. Cross-architecture pools models across architectures, holding the dataset and optimizer fixed. Cross-optimizer & cross-architecture pools models across architectures and optimizers. Cross-dataset & cross-architecture pools models across architectures and datasets. See details in Appendix C.

(S, C) Metric Pair	Cross-architecture		Cross-optimizer & Cross-architecture		Cross-dataset & Cross-architecture			
	PCR	R^2 (S, C)	PCR	R^2 (S, C)	PCR	R^2 (S, C)	PCR	R^2 (S, C)
$(\mathrm{trace}(\mathrm{H})$, func norm$)$	4.1%	0.81 $(+,+)$	11.5%	0.26 $(-,+)$	2.3%	0.83 $(+,+)$	2.2%	0.86 $(+,+)$
$(\lambda_{\max}(\mathrm{H})$, func norm$)$	4.1%	0.81 $(+,+)$	11.6%	0.25 $(-,+)$	2.2%	0.81 $(+,+)$	2.9%	0.86 $(+,+)$
$(\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$, func norm$)$	2.1%	0.88 $(+,+)$	6.6%	0.32 $(+,+)$	3.0%	0.92 $(+,+)$	2.7%	0.87 $(+,+)$
$(\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$, func norm$)$	1.7%	0.89 $(+,+)$	6.1%	0.25 $(+,+)$	2.4%	0.92 $(+,+)$	2.4%	0.87 $(+,+)$
$(\mathrm{trace}(\mathrm{H})$, param norm$)$	65.6%	0.18 $(-,+)$	64.6%	0.01 $(-,-)$	8.5%	0.81 $(+,+)$	41.0%	0.05 $(+,+)$
$(\lambda_{\max}(\mathrm{H})$, param norm$)$	62.3%	0.17 $(-,+)$	63.5%	0.01 $(-,-)$	9.8%	0.81 $(+,+)$	39.8%	0.03 $(+,+)$
$(\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$, param norm$)$	29.9%	0.17 $(+,+)$	27.0%	0.14 $(+,-)$	9.2%	0.89 $(+,+)$	30.7%	0.04 $(+,-)$
$(\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$, param norm$)$	27.7%	0.19 $(+,+)$	26.3%	0.05 $(+,-)$	8.4%	0.88 $(+,+)$	27.8%	0.11 $(+,-)$
$(\mathrm{trace}(\mathrm{H})$, path norm$)$	70.1%	0.11 $(-,+)$	49.4%	0.61 $(-,+)$	28.2%	0.35 $(-,+)$	51.9%	0.04 $(+,-)$
$(\lambda_{\max}(\mathrm{H})$, path norm$)$	68.5%	0.10 $(-,+)$	49.1%	0.61 $(-,+)$	30.8%	0.35 $(-,+)$	51.6%	0.02 $(+,-)$
$(\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$, path norm$)$	36.8%	0.10 $(+,+)$	25.8%	0.61 $(+,+)$	17.1%	0.75 $(+,+)$	36.1%	0.05 $(+,-)$
$(\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$, path norm$)$	36.8%	0.11 $(+,+)$	25.2%	0.61 $(+,+)$	17.9%	0.67 $(+,+)$	34.2%	0.12 $(+,-)$
$(\mathrm{trace}(\mathrm{H})$, spec norm$)$	66.8%	0.11 $(-,+)$	47.5%	0.26 $(-,+)$	43.4%	0.11 $(-,+)$	53.3%	0.04 $(+,+)$
$(\lambda_{\max}(\mathrm{H})$, spec norm$)$	64.5%	0.11 $(-,+)$	47.2%	0.26 $(-,+)$	47.5%	0.11 $(-,+)$	53.0%	0.02 $(+,+)$
$(\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}$, spec norm$)$	28.2%	0.12 $(+,+)$	19.6%	0.35 $(+,+)$	14.8%	0.70 $(+,+)$	32.7%	0.04 $(+,+)$
$(\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}$, spec norm$)$	29.7%	0.14 $(+,+)$	19.3%	0.30 $(+,+)$	20.1%	0.61 $(+,+)$	31.4%	0.11 $(+,+)$

6 Conclusion

Focusing on the central question posed in the paper's title, we quantitatively examined the reach of the two-factor view in modern deep networks. In contrast to the pessimistic findings of previous studies on scalar generalization measures, including scalarized sharpness-complexity measures, our results show that suitably defined (S, C) metric pairs can provide strong explanatory power across a range of settings. We find that finite-scale sharpness often exhibits much stronger explanatory power than its infinitesimal counterpart, with perturbation scale typically mattering more than the Hessian statistic choice. Function norm offers more stable explanatory behavior than parameter norm in many settings, particularly in models without normalization and in mixed settings. Despite being scale-variant, parameter norm remains highly effective in some training regimes, highlighting that the advantage of complexity representation depends on the setting. This observation is consistent with the practical effectiveness of weight decay in SGD training. The remaining failures are also informative. An important next step is to investigate whether the unexplained regimes call for more refined metrics or for additional factor(s) beyond the two factors. We hope these findings can guide further investigation toward a more comprehensive understanding of generalization.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing, format references, and assist with writing and debugging code. We did not use generative AI tools to develop our main conceptual framework, propose the methodology, select the metrics investigated, or interpret the results. Synthetic data generation is not applicable to this work.

We have reviewed all AI-assisted work. We checked the revised text and reference formatting. We tested the AI-generated code and verified its correctness by having different people independently repeat the experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Pierre Alquier. User-friendly introduction to pac-bayes bounds. *Foundations and Trends in Machine Learning*, 17(2):174–303, 2024.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, 2022.

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 840–902. PMLR, 2023.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 254–263. PMLR, 2018.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Peter L Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

Peter L. Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: A statistical viewpoint. *Acta Numerica*, 30:87–201, 2021.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017.

Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning? In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028, 2017.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017.

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M. Roy. In search of robust measures of generalization. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in PAC-Bayes bounds. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130, pp. 604–612. PMLR, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241. PMLR, 2019.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *Information and Inference: A Journal of the IMA*, 9(2):473–501, 2020.

Benjamin Guedj. A primer on pac-bayesian learning. *arXiv preprint arXiv:1901.05353*, 2019.

Suriya Gunasekar, Jason D. Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

Simran Kaur, Jeremy M. Cohen, and Zachary C. Lipton. On the maximum hessian eigenvalue and generalization. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2023.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, volume 4, 1991.

Jungmin Kwon, Jeongmin Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, 2021.

Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4): 255–282, 1950.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-Rao metric, geometry, and complexity of neural networks. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 888–896. PMLR, 2019.

Yucong Liu, Shixing Yu, and Tong Lin. Regularizing deep neural networks with stochastic estimators of hessian trace. *arXiv preprint arXiv:2208.05924*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 230–234, 1998.

David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170, 1999.

David A. McAllester. Simplified PAC-Bayesian margin bounds. In *Learning Theory and Kernel Machines (COLT)*, 2003.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

Henning Petzka, Michael Kamp, Linus Schäfer, Asif Salam, and Christian Fischer. Relative flatness and generalization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10420–10432, 2021.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70): 1–57, 2018.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using PAC-Bayesian analysis. In *International Conference on Machine Learning*, pp. 9636–9647. PMLR, 2020.

Colin Wei and Tengyu Ma. Improved sample complexities for deep networks and robust classification via an all-layer margin. In *International Conference on Learning Representations*, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2019.

APPENDIX

A THEORETICAL MOTIVATION OF THE TWO-FACTOR VIEW

In this section, we explain the theoretical motivation of the two-factor view.

A.1 PAC-BAYES BOUNDS

PAC-Bayes bounds theoretically characterize the generalization gap in terms of sharpness and complexity. A typical PAC-Bayes bound relates the test performance of a randomized predictor to its training performance and complexity; see Section 3.2.2 in Alquier (2024).

Let $Z = \{z_i\}_{i=1}^m$ be a training set containing m independent samples drawn from the data distribution $\mathcal{D}$. Let $\mathcal{P}$ be a prior distribution over predictors, chosen independently of Z, and let $\mathcal{Q}$ be a posterior distribution that may depend on Z. Assuming that the loss is bounded in $[0, 1]$, with probability at least $1 - \delta$ over the random draw of Z, the following inequality holds for all posterior distributions $\mathcal{Q}$:

$$L_{\mathcal{D}}(\mathcal{Q}) \quad \leq \quad \widehat{L}_Z(\mathcal{Q}) + \sqrt{\frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\left(\frac{2\sqrt{m}}{\delta}\right)}{2m}}. \tag{6}$$

Here,

$$L_{\mathcal{D}}(\mathcal{Q}) := \mathbb{E}_{h\sim\mathcal{Q}}\ell(h;\mathcal{D}) = \mathbb{E}_{h\sim\mathcal{Q}}\mathbb{E}_{z\sim\mathcal{D}}\big[\ell(h;z)\big], \tag{7}$$

is the *expected* population loss (or test loss) of a predictor h sampled from $\mathcal{Q}$. Similarly,

$$\widehat{L}_Z(\mathcal{Q}) := \mathbb{E}_{h\sim\mathcal{Q}}\big[\ell(h;Z)\big] = \mathbb{E}_{h\sim\mathcal{Q}}\Big[\frac{1}{m}\sum\nolimits_{i=1}^{m}\ell(h;z_i)\Big] \tag{8}$$

is the *expected* empirical loss of a predictor h sampled from $\mathcal{Q}$.

The KL divergence $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ measures how different the posterior distribution $\mathcal{Q}$, which represents the trained predictors, is from the data-independent prior distribution $\mathcal{P}$. It therefore serves as a measure of model complexity.

In simple terms, the PAC-Bayes bound 6 states that the *test loss is small when two quantities are small*: the expected empirical loss $\widehat{L}_Z(\mathcal{Q})$ and the complexity term $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$. The bound also becomes tighter as the number of training samples m becomes larger.

As mentioned in Section 3.1, for deep networks, the model h is usually represented through parameterization, i.e., $h = f_w$. This leads to two levels of description. In **parameter space**, we denote the prior and posterior distributions by $\mathcal{P}_w$ and $\mathcal{Q}_w$. Through the map $w \mapsto f_w$, they induce corresponding **function space** distributions, denoted by $\mathcal{P}_f$ and $\mathcal{Q}_f$.

A.2 ORIGIN OF THE TWO FACTORS

PAC-Bayes bounds control the expected population loss $L_{\mathcal{D}}(\mathcal{Q})$ mainly through two terms:

the expected empirical loss $\hat{L}_Z(\mathcal{Q})$ and *the posterior-prior deviation* $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$.

The first term is closely related to the **sharpness** factor.

Applying Taylor expansion in the parameter space to $\hat{L}_Z(\mathcal{Q})$, we obtain

$$\widehat{L}_Z(\mathcal{Q}_w) \;=\; \mathbb{E}_{w\sim\mathcal{Q}_w}\big[\ell(f_w;Z)\big] \;= \ell(f_{\hat{w}};Z) + \tfrac{\sigma^2}{2}\,\mathrm{Tr}\left(\nabla_w^2\ell(f_{\hat{w}};Z)\right) + o(\sigma^2), \tag{9}$$

where the posterior $\mathcal{Q}_w$ is centered at the learned solution $\hat{w}$ with covariance $\Sigma_w = \sigma^2 I$; see the proof below. This shows that, when the training loss $\ell(f_{\hat{w}};Z)$ becomes sufficiently small, the dominant term depends on the *local curvature* around the learned solution $\hat{w}$, which provides intuition of the sharpness factor.

The second term is closely related to the **complexity** factor: KL divergence $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ measures how much deviation is needed to move from the prior to the learned posterior.

KL divergence admits a closed form under a Gaussian posterior and prior. Suppose $\mathcal{Q} = \mathcal{N}(\mu_{\mathcal{Q}}, \Sigma_{\mathcal{Q}})$ and $\mathcal{P} = \mathcal{N}(\mu_{\mathcal{P}}, \Sigma_{\mathcal{P}})$,

$$\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) = \frac{1}{2}\Big(-d + \mathrm{Tr}\big(\Sigma_{\mathcal{Q}}\ {\Sigma_{\mathcal{P}}}^{-1}\big) + \ln\frac{\det(\Sigma_{\mathcal{P}})}{\det(\Sigma_{\mathcal{Q}})} + (\mu_{\mathcal{Q}} - \mu_{\mathcal{P}})^\top {\Sigma_{\mathcal{P}}}^{-1}(\mu_{\mathcal{Q}} - \mu_{\mathcal{P}})\Big)\ , \quad (10)$$

where d is the dimension of the distribution.

In particular, assume the parameter-level prior and posterior are Gaussian with same isotropic covariance $\Sigma_{\mathcal{Q}_w} = \Sigma_{\mathcal{P}_w} = \sigma^2 I$, posterior is centered at the learned solution $\mu_{\mathcal{Q}_w} = \hat{w}$ and prior is centered at the origin $\mu_{\mathcal{P}_w} = 0$, then the parameter-level KL divergence reduces to squared weight norm up to a constant factor

$$\mathrm{KL}(\mathcal{Q}_w\|\mathcal{P}_w) = \frac{\|\hat{w}\|^2}{2\sigma^2}. \quad (11)$$

This corresponds to the *weight decay* in SGD training, and thus explains why weight decay can be viewed as a simple parameter-space complexity regularizer.

Proof. Let the posterior $\mathcal{Q}_w$ be centered at the learned solution $\hat{w}$ with covariance $\Sigma_w = \sigma^2 I$, the expected empirical loss can be written as

$$\hat{L}_Z(\mathcal{Q}_w) := \mathbb{E}_{w\sim\mathcal{Q}_w}\big[\ell(f_w; Z)\big] \overset{w=\hat{w}+\Delta w}{=} \mathbb{E}_{\Delta w\sim\widetilde{\mathcal{Q}}_w}\big[\ell(f_{\hat{w}+\Delta w}; Z)\big],$$

where $\widetilde{\mathcal{Q}}_w$ is the distribution of $\Delta w = w - \hat{w}$, satisfying

$$\mathbb{E}_{\widetilde{\mathcal{Q}}_w}[\Delta w] = 0\ , \quad \mathbb{E}_{\widetilde{\mathcal{Q}}_w}[\Delta w \Delta w^\top] = \sigma^2 I.$$

Applying a second-order Taylor expansion around $\hat{w}$, we obtain

$$\begin{aligned}
\hat{L}_Z(\mathcal{Q}_w) &= \mathbb{E}_{\Delta w\sim\widetilde{\mathcal{Q}}_w}\Big[\ell(f_{\hat{w}+\Delta w}; Z)\Big] \\
&= \mathbb{E}_{\Delta w\sim\widetilde{\mathcal{Q}}_w}\Big[\ell(f_{\hat{w}}; Z) + \Delta w^\top \nabla_w \ell(f_{\hat{w}}; Z) + \tfrac{1}{2}\Delta w^\top \nabla_w^2 \ell(f_{\hat{w}}; Z)\Delta w + o(\|\Delta w\|^2)\Big] \\
&= \ell(f_{\hat{w}}; Z) + 0 + \tfrac{1}{2}\,\mathrm{Tr}\Big(\nabla_w^2 \ell(f_{\hat{w}}; Z)\,\mathbb{E}_{\Delta w\sim\widetilde{\mathcal{Q}}_w}[\Delta w \Delta w^\top]\Big) + \mathbb{E}_{\Delta w\sim\widetilde{\mathcal{Q}}_w}[o(\|\Delta w\|^2)] \\
&= \underbrace{\ell(f_{\hat{w}}; Z)}_{\text{training loss}} + \quad \tfrac{\sigma^2}{2}\,\mathrm{Tr}\Big(\nabla_w^2 \ell(f_{\hat{w}}; Z)\Big) + o(\sigma^2).
\end{aligned}$$

□

A.3 Finite-scale sharpness

For convenience, denote $T := \mathrm{diag}(|w|)$, and we have the adaptive Hessian

$$H = T\nabla_w^2 \ell(f_w; Z)T.$$

Finite-scale Hessian trace. As introduced in Section 3, the finite-scale Hessian trace is

$$\mathrm{trace(H)}_{\mathrm{fs}}(w; \sigma) := \mathbb{E}_\epsilon\left[\frac{\ell(f_{w+\sigma\epsilon}; Z) - \ell(f_w; Z)}{\sigma^2}\right], \qquad \epsilon = Tz, \quad z \sim \mathcal{N}(0, I). \quad (12)$$

We show that this quantity approaches one-half of the infinitesimal adaptive Hessian trace as the perturbation scale approaches zero.

Proof. Let $\Delta w = \sigma T z$. Then

$$\mathbb{E}[\Delta w] = 0, \qquad \mathbb{E}[\Delta w \Delta w^\top] = \sigma^2 T^2.$$

Taking the expectation of the Taylor expansion, we obtain

$$\begin{aligned}
\mathbb{E}_{\Delta w}\big[\ell(f_{w+\Delta w}; Z)\big] &= \mathbb{E}_{\Delta w}\Big[\ell(f_w; Z) + \Delta w^\top \nabla_w \ell(f_w; Z) + \tfrac{1}{2}\Delta w^\top \nabla_w^2 \ell(f_w; Z)\Delta w + o(\|\Delta w\|_2^2)\Big] \\
&= \ell(f_w; Z) + 0 + \tfrac{1}{2}\,\mathrm{tr}\left(\nabla_w^2 \ell(f_w; Z)\,\mathbb{E}[\Delta w \Delta w^\top]\right) + o(\sigma^2) \\
&= \ell(f_w; Z) + \tfrac{\sigma^2}{2}\,\mathrm{tr}\left(\nabla_w^2 \ell(f_w; Z) T^2\right) + o(\sigma^2) \\
&= \ell(f_w; Z) + \tfrac{\sigma^2}{2}\,\mathrm{tr}(H) + o(\sigma^2).
\end{aligned}$$

Subtracting $\ell(f_w; Z)$ and dividing by σ^2 yields

$$\mathrm{trace}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) = \tfrac{1}{2}\,\mathrm{tr}(H) + o(1) \;\longrightarrow\; \tfrac{1}{2}\,\mathrm{tr}(H) \qquad \text{as } \sigma \to 0.$$

Thus, up to a constant factor, the finite-scale Hessian trace approaches its infinitesimal counterpart. □

Finite-scale Hessian top eigenvalue. The finite-scale Hessian top eigenvalue is

$$\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) := \max_{\|v\|_2=1} \frac{\ell(f_{w+\sigma Tv}; Z) + \ell(f_{w-\sigma Tv}; Z) - 2\ell(f_w; Z)}{\sigma^2}. \tag{13}$$

We show that this quantity approaches the infinitesimal adaptive Hessian top eigenvalue as the perturbation scale approaches zero.

Proof. For any unit vector v, a second-order Taylor expansion gives

$$\ell(f_{w\pm\sigma Tv}; Z) = \ell(f_w; Z) \pm \sigma v^\top T \nabla_w \ell(f_w; Z) + \tfrac{\sigma^2}{2} v^\top H v + o(\sigma^2).$$

Adding the two expansions cancels the linear terms, yielding

$$\frac{\ell(f_{w+\sigma Tv}; Z) + \ell(f_{w-\sigma Tv}; Z) - 2\ell(f_w; Z)}{\sigma^2} = v^\top H v + o(1).$$

Taking the maximum over unit vectors and using the Rayleigh characterization, we obtain

$$\begin{aligned}\lambda_{\max}(\mathrm{H})_{\mathrm{fs}}(w;\sigma) &= \max_{\|v\|_2=1} v^\top H v + o(1) \\ &= \lambda_{\max}(H) + o(1) \;\longrightarrow\; \lambda_{\max}(H) \qquad \text{as } \sigma \to 0.\end{aligned}$$

□

A.4 Further interpretations on the function norm

The complexity term $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ in PAC-Bayes bounds is fundamentally *defined over distributions of predictors* rather than over a particular parameter representation. However, in deep networks, the prior and posterior are commonly specified in parameter space, leading to the parameter-level complexity $\mathrm{KL}(\mathcal{Q}_\theta\|\mathcal{P}_\theta)$. Because the map from parameters to predictors $\theta \mapsto f_\theta$ is generally non-injective in deep networks, this quantity may distinguish parameter distributions that induce the same predictive behavior.

To obtain a complexity metric defined closer to function space, we map parameters to their centered function outputs on a fixed reference dataset $D_{\mathrm{ref}} = \{\mathbf{x}_i\}_{i=1}^n$. Let Z_w denote the logit matrix:

$$Z_w(D_{\mathrm{ref}}) := \begin{bmatrix} f_w(\mathbf{x}_1)^\top \\ \vdots \\ f_w(\mathbf{x}_n)^\top \end{bmatrix} \in \mathbb{R}^{n\times c},$$

where $f_w(\mathbf{x}_i)$ is the c-class logit vector for $\mathbf{x}_i$. Softmax probabilities are invariant under the transformation $Z_w \mapsto Z_w + \mathbf{e}\mathbf{1}_c^\top$ for any $\mathbf{e} \in \mathbb{R}^n$. We remove this additive ambiguity using the class-centering projection $\Pi_c = I_c - \frac{1}{c}\mathbf{1}_c\mathbf{1}_c^\top$. The *class-centered logit matrix* is then defined as

$$\widetilde{Z}_w(D_{\mathrm{ref}}) := Z_w(D_{\mathrm{ref}})\Pi_c = \begin{bmatrix} \widetilde{f}_w(\mathbf{x}_1)^\top \\ \vdots \\ \widetilde{f}_w(\mathbf{x}_n)^\top \end{bmatrix} \in \mathbb{R}^{n\times c},$$

where $\widetilde{f}_w(\mathbf{x}_i) = f_w(\mathbf{x}_i) - \frac{1}{c}\mathbf{1}_c\mathbf{1}_c^\top f_w(\mathbf{x}_i)$.

Define the centered function-output lifting map as

$$\widetilde{F}_{D_{\mathrm{ref}}}(w) := \mathrm{vec}\left(\widetilde{Z}_w(D_{\mathrm{ref}})\right) = \mathrm{vec}\left(\begin{bmatrix} \widetilde{f}_w(x_1)^\top \\ \vdots \\ \widetilde{f}_w(x_n)^\top \end{bmatrix}\right). \tag{14}$$

Through the lifting map in equation 14, the parameter-space prior and posterior induce corresponding distributions over the centered function-outputs:

$$\mathcal{P}_f^{D_{\mathrm{ref}}} := \left(\widetilde{F}_{D_{\mathrm{ref}}}\right)_{\#} \mathcal{P}_w, \qquad \mathcal{Q}_f^{D_{\mathrm{ref}}} := \left(\widetilde{F}_{D_{\mathrm{ref}}}\right)_{\#} \mathcal{Q}_w. \tag{15}$$

Therefore, a natural **function-space counterpart** of the parameter-space complexity is

$$\mathrm{KL}\Big(\mathcal{Q}_f^{D_{\mathrm{ref}}} \,\|\, \mathcal{P}_f^{D_{\mathrm{ref}}}\Big). \tag{16}$$

Because the centered function-output distributions are obtained by applying a deterministic map to the parameter distributions, the data-processing inequality gives

$$\mathrm{KL}\Big(\mathcal{Q}_f^{D_{\mathrm{ref}}} \,\|\, \mathcal{P}_f^{D_{\mathrm{ref}}}\Big) \le \mathrm{KL}(\mathcal{Q}_w \,\|\, \mathcal{P}_w).$$

The function-space KL divergence 16 *discards distinctions between parameter distributions* that disappear after mapping them to the same predictor on the reference inputs.

Directly estimating the function-level KL divergence 16 can be computationally demanding. We therefore consider a *simplified proxy*. The centered function outputs lie in the subspace

$$\mathcal{H}_{D_{\mathrm{ref}}} := \left\{\mathrm{vec}(U) : U \in \mathbb{R}^{n\times c},\ U\mathbf{1}_c = 0\right\}.$$

Suppose that the induced function-space posterior and prior are approximated on this subspace by Gaussian distributions with a shared isotropic covariance, and prior is zero-centered, denoted by

$$\mathcal{Q}_f^{D_{\mathrm{ref}}} \approx \mathcal{N}_{\mathcal{H}_{D_{\mathrm{ref}}}}\left(\mu_f, \sigma_f^2 I\right), \qquad \mathcal{P}_f^{D_{\mathrm{ref}}} \approx \mathcal{N}_{\mathcal{H}_{D_{\mathrm{ref}}}}\left(0, \sigma_f^2 I\right).$$

Under these assumptions, according to Eq. 10, the function-space KL divergence 16 reduces to

$$\mathrm{KL}\Big(\mathcal{Q}_f^{D_{\mathrm{ref}}} \,\|\, \mathcal{P}_f^{D_{\mathrm{ref}}}\Big) = \frac{\|\mu_f\|_2^2}{2\sigma_f^2}. \tag{17}$$

When the parameter posterior is centered at the learned solution (i.e., $\mathcal{Q}_w$ has mean $\hat{w}$), its induced function-level posterior mean can be approximated by the centered function-outputs at the trained model:

$$\mu_f = \mathbb{E}_{w\sim\mathcal{Q}_w}\left[\widetilde{F}_{D_{\mathrm{ref}}}(w)\right] \approx \widetilde{F}_{D_{\mathrm{ref}}}(\hat{w}).$$

As a result,

$$\mathrm{KL}\Big(\mathcal{Q}_f^{D_{\mathrm{ref}}} \,\|\, \mathcal{P}_f^{D_{\mathrm{ref}}}\Big) \approx \frac{1}{2\sigma_f^2}\|\widetilde{F}_{D_{\mathrm{ref}}}(\hat{w})\|_2^2. \tag{18}$$

Recall that the **function norm** in Section 3.2 is defined as

$$\text{func norm} := \left\|\widetilde{Z}_w(D_{\mathrm{ref}})\right\|_F,$$

which is the unsquared norm appearing in 18, with the constant factor omitted. The fixed factor $1/2\sigma_f^2$ can be omitted because it does not affect Pareto analysis or linear regression results.

This centered function-output norm can be viewed as a practical ***proxy*** for a simplified function-space posterior–prior KL divergence. It is not an exact evaluation of 16; rather, it is obtained under the approximations of a shared isotropic function-space covariance, a zero-centered function prior, and a concentrated posterior.

B Detailed explanation of the diagnostic criteria

Pareto analysis and linear regression provide complementary diagnostics of the explanatory power of an evaluated metric pair.

PCR measures violations of the monotonic ordering predicted by the two-factor view (*lower is better*), while R^2 evaluates how well a linear model captures the relationship (*higher is better*).

The possible outcomes of interpreting their diagnostics jointly are summarized in Table 3 and discussed below.

i. *Low PCR, high R^2, and positive regression coefficients.* This is the ideal case. The metric pair induces an ordering that agrees with the generalization ordering, and the relationship is well captured by a linear model. Together, these diagnostics provide strong evidence that the evaluated metric pair has strong explanatory power for generalization.

ii. *Low PCR, low R^2, and positive regression coefficients.* The metric pair induces an ordering that is consistent with the two-factor view, but the relationship is not well captured by a linear model. In this case, poor regression diagnostics do not necessarily weaken the two-factor view, which may retain strong explanatory power even when the dependence of generalization on the evaluated metrics is nonlinear.

iii. *High PCR, high R^2, and positive regression coefficients.* The regression captures a strong global linear trend, but the high PCR reveals frequent violations of the expected monotonic ordering. Because R^2 describes the quality of an *approximate* global linear fit, small residuals can violate the ordering of nearby models and worsen PCR while barely affecting R^2. Thus, high R^2 does not guarantee low PCR. In this case, the linear trend captures part of the relationship, while important information may still be missing.

iv. *High PCR and low R^2, or any negative regression coefficient.* High PCR together with low R^2 indicates that the metric pair is neither order-consistent nor well described by a linear model. A negative regression coefficient constitutes an additional sign failure, because it contradicts the expected direction of the corresponding sharpness or complexity effect. This indicates that the evaluated metrics are poor realizations of the two factors in the given setting.

Greater emphasis on PCR.

The joint interpretation *places greater weight on* PCR than on R^2, because PCR does not rely on a linear modeling assumption. For example, a diagnostic result with low PCR but low R^2 is still considered favorable: it suggests strong agreement with the monotonic ordering, even though the relationship is not well captured by a linear model.

Why retain linear regression diagnostics when PCR takes priority?

Although we prioritize PCR when evaluating explanatory power, linear regression provides additional information about the form of the relationship beyond pairwise ordering. This may help guide regularization, where selected quantities are often added to the training objective as a weighted sum; see the distinction of regularization versus explanation in Section 2. Low PCR together with high R^2 and positive coefficients supports a simple additive approximation, motivating the corresponding weighted sum as a promising candidate training penalty. With low PCR but a poor linear fit, the monotonic ordering remains informative even though a fixed weighted sum does not adequately capture the relationship. Therefore linear regression *plays a complementary role*: it helps assess how the two factors might be combined, providing additional information that PCR alone cannot supply.

C Experimental protocol

We report several implementation choices that are important for conducting the experiments.

1. **Broad hyperparameter sweeps.** For each individual setting in Table 4 and Table 7, we perform a broad hyperparameter sweep over learning rate, batch size, weight decay, and momentum, and obtain at least 48 trained models. The specific hyperparameter ranges may differ across architectures to account for differences in training stability and optimization behavior.

2. **Filtering under-trained models.** To avoid confounding effects from under-trained models, we retain well-trained models that achieve sufficiently small training loss, using a threshold of $\text{train loss} \leq 0.01$. Runs that do not satisfy this criterion are not used in the diagnostic analyses. This filtering step helps ensure that the measured generalization gap primarily reflects the variations in generalization, rather than artifacts of incomplete training or optimization failure. For each individual setting in Table 4 and Table 7, we retain at least 24 well-trained models for diagnostic analyses. This ensures that the Pareto analysis and linear regression are conducted on a diverse collection of well-trained models.
3. **Fixed reference data for metric computation.** The complexity metric, function norm, requires reference data. It is computed using the same fixed reference set for all trained models. This reduces sampling noise and ensures that differences in metric values are attributable to the trained models, rather than to variations in the reference data used for evaluation.

Table 6 summarizes the counts of trained models and well-trained models that are retained in each individual setting.

Table 6: Counts of trained models and retained well-trained models in each individual setting. For the last column, the denominator denotes the total number of trained models obtained from the hyperparameter sweep, and the numerator denotes the number of well-trained models that are retained for analysis (train loss ≤ 0.01).

Dataset	Architecture	Optimizer	Retained / Total
CIFAR-10	ResNet18	SGD	30 / 48
	VGG13	SGD	34 / 48
	ResNet18 no-BN	SGD	58 / 60
	VGG13 no-BN	SGD	30 / 50
	ResNet18	AdamW	46 / 48
	VGG13	AdamW	39 / 48
	ResNet18 no-BN	AdamW	35 / 48
	VGG13 no-BN	AdamW	37 / 48
CIFAR-100	WideResNet	SGD	24 / 48
	ViT	AdamW	47 / 48
	WideResNet no-BN	SGD	47 / 48
	ViT no-LN	AdamW	28 / 48

D ALL NUMERICAL RESULTS

D.1 OPTIMIZER ADAMW

Table 7: **Main results**. Experimental settings, evaluated metric pairs and diagnostic criteria are the same as in Table 4. Here, models are trained using AdamW, rather than SGD.

(S, C) Metric Pair	CIFAR-10							
	ResNet18 adamw		**VGG13** adamw		**ResNet18 no-BN** adamw		**VGG13 no-BN** adamw	
	PCR	$R^2\,(S,C)$	**PCR**	$R^2\,(S,C)$	**PCR**	$R^2\,(S,C)$	**PCR**	$R^2\,(S,C)$
(trace(H) , func norm)	44.1%	0.59 (+, +)	18.0%	0.83 (+, +)	32.5%	0.69 (−, +)	28.4%	0.15 (+, +)
($\lambda_{\max}$(H) , func norm)	46.1%	0.59 (+, +)	17.1%	0.83 (+, +)	27.5%	0.67 (−, +)	29.9%	0.15 (+, +)
($trace(H)_{fs}$, func norm)	3.6%	0.70 (+, +)	6.7%	0.91 (+, +)	6.0%	0.75 (+, +)	6.0%	0.78 (+, +)
($\lambda_{max}(H)_{fs}$, func norm)	4.3%	0.64 (+, +)	5.5%	0.81 (+, +)	6.7%	0.73 (+, +)	10.8%	0.66 (+, +)
(trace(H) , param norm)	28.9%	0.78 (+, +)	8.8%	0.57 (+, +)	42.5%	0.51 (−, +)	17.0%	0.79 (+, +)
($\lambda_{\max}$(H) , param norm)	30.1%	0.78 (+, +)	8.1%	0.57 (+, +)	36.7%	0.49 (−, +)	17.0%	0.79 (+, +)
($trace(H)_{fs}$, param norm)	10.1%	0.76 (+, +)	7.3%	0.91 (+, +)	10.5%	0.43 (−, +)	4.5%	0.85 (+, +)
($\lambda_{max}(H)_{fs}$, param norm)	9.7%	0.78 (+, +)	7.1%	0.81 (+, +)	9.6%	0.51 (−, +)	5.9%	0.93 (+, +)
(trace(H) , path norm)	56.2%	0.48 (−, +)	31.0%	0.70 (−, +)	43.9%	0.59 (−, +)	16.6%	0.73 (+, +)
($\lambda_{\max}$(H) , path norm)	56.7%	0.48 (−, +)	29.5%	0.70 (−, +)	37.9%	0.56 (−, +)	16.5%	0.73 (+, +)
($trace(H)_{fs}$, path norm)	29.3%	0.60 (−, +)	19.6%	0.89 (+, −)	10.0%	0.55 (+, +)	4.6%	0.84 (+, +)
($\lambda_{max}(H)_{fs}$, path norm)	28.3%	0.48 (−, +)	17.9%	0.89 (+, +)	9.5%	0.56 (−, +)	6.1%	0.91 (+, +)
(trace(H) , spec norm)	42.2%	0.64 (+, +)	17.8%	0.66 (+, +)	44.9%	0.56 (−, +)	16.1%	0.66 (+, +)
($\lambda_{\max}$(H) , spec norm)	44.1%	0.64 (+, +)	15.9%	0.66 (+, +)	38.3%	0.52 (−, +)	16.3%	0.66 (+, +)
($trace(H)_{fs}$, spec norm)	12.7%	0.69 (+, +)	8.9%	0.94 (+, +)	9.7%	0.49 (+, +)	4.7%	0.83 (+, +)
($\lambda_{max}(H)_{fs}$, spec norm)	12.6%	0.69 (+, +)	8.7%	0.87 (+, +)	9.4%	0.49 (−, +)	6.0%	0.89 (+, +)

D.2 MIXED SETTINGS

Table 8: A detailed version of Table 5. The last row lists models being pooled in each mixed setting, with the well-trained model counts in parentheses.

(S, C) Metric Pair	**Cross-architecture**		**Cross-optimizer & Cross-architecture**		**Cross-dataset & Cross-architecture**			
	PCR	$R^2\,(S,C)$	**PCR**	$R^2\,(S,C)$	**PCR**	$R^2\,(S,C)$	**PCR**	$R^2\,(S,C)$
(trace(H) , func norm)	4.1%	0.81 (+, +)	11.5%	0.26 (−, +)	2.3%	0.83 (+, +)	2.2%	0.86 (+, +)
($\lambda_{\max}$(H) , func norm)	4.1%	0.81 (+, +)	11.6%	0.25 (−, +)	2.2%	0.81 (+, +)	2.9%	0.86 (+, +)
($\text{trace(H)}_{\text{fs}}$, func norm)	2.1%	0.88 (+, +)	6.6%	0.32 (+, +)	3.0%	0.92 (+, +)	2.7%	0.87 (+, +)
($\lambda_{\max}(\text{H})_{\text{fs}}$, func norm)	1.7%	0.89 (+, +)	6.1%	0.25 (+, +)	2.4%	0.92 (+, +)	2.4%	0.87 (+, +)
(trace(H) , param norm)	65.6%	0.18 (−, +)	64.6%	0.01 (−, −)	8.5%	0.81 (+, +)	41.0%	0.05 (+, +)
($\lambda_{\max}$(H) , param norm)	62.3%	0.17 (−, +)	63.5%	0.01 (−, −)	9.8%	0.81 (+, +)	39.8%	0.03 (+, +)
($\text{trace(H)}_{\text{fs}}$, param norm)	29.9%	0.17 (+, +)	27.0%	0.14 (+, −)	9.2%	0.89 (+, +)	30.7%	0.04 (+, −)
($\lambda_{\max}(\text{H})_{\text{fs}}$, param norm)	27.7%	0.19 (+, +)	26.3%	0.05 (+, −)	8.4%	0.88 (+, +)	27.8%	0.11 (+, −)
(trace(H) , path norm)	70.1%	0.11 (−, +)	49.4%	0.61 (−, +)	28.2%	0.35 (−, +)	51.9%	0.04 (+, −)
($\lambda_{\max}$(H) , path norm)	68.5%	0.10 (−, +)	49.1%	0.61 (−, +)	30.8%	0.35 (−, +)	51.6%	0.02 (+, −)
($\text{trace(H)}_{\text{fs}}$, path norm)	36.8%	0.10 (+, +)	25.8%	0.61 (+, +)	17.1%	0.75 (+, +)	36.1%	0.05 (+, −)
($\lambda_{\max}(\text{H})_{\text{fs}}$, path norm)	36.8%	0.11 (+, +)	25.2%	0.61 (+, +)	17.9%	0.67 (+, +)	34.2%	0.12 (+, −)
(trace(H) , spec norm)	66.8%	0.11 (−, +)	47.5%	0.26 (−, +)	43.4%	0.11 (−, +)	53.3%	0.04 (+, +)
($\lambda_{\max}$(H) , spec norm)	64.5%	0.11 (−, +)	47.2%	0.26 (−, +)	47.5%	0.11 (−, +)	53.0%	0.02 (+, +)
($\text{trace(H)}_{\text{fs}}$, spec norm)	28.2%	0.12 (+, +)	19.6%	0.35 (+, +)	14.8%	0.70 (+, +)	32.7%	0.04 (+, +)
($\lambda_{\max}(\text{H})_{\text{fs}}$, spec norm)	29.7%	0.14 (+, +)	19.3%	0.30 (+, +)	20.1%	0.61 (+, +)	31.4%	0.11 (+, +)
Models being pooled together (the number of well-trained models in the collection)	CIFAR-10, sgd ResNet18 (30) VGG13 (34) ResNet18 no-BN (58) VGG13 no-BN (30)		CIFAR-10, sgd&adamw ResNet18 (76) VGG13 (73) ResNet18 no-BN (93) VGG13 no-BN (67)		CIFAR-10, sgd ResNet18 (30) VGG13 (34) CIFAR-100, sgd WideResNet (24)		CIFAR-10, sgd ResNet18 (30) VGG13 (34) ResNet18 no-BN (58) VGG13 no-BN (30) CIFAR-100, sgd WideResNet (24) WideResNet no-BN (47)	
Collection size	152		309		88		223	

D.3 VARYING SHARPNESS PERTURBATION SCALES

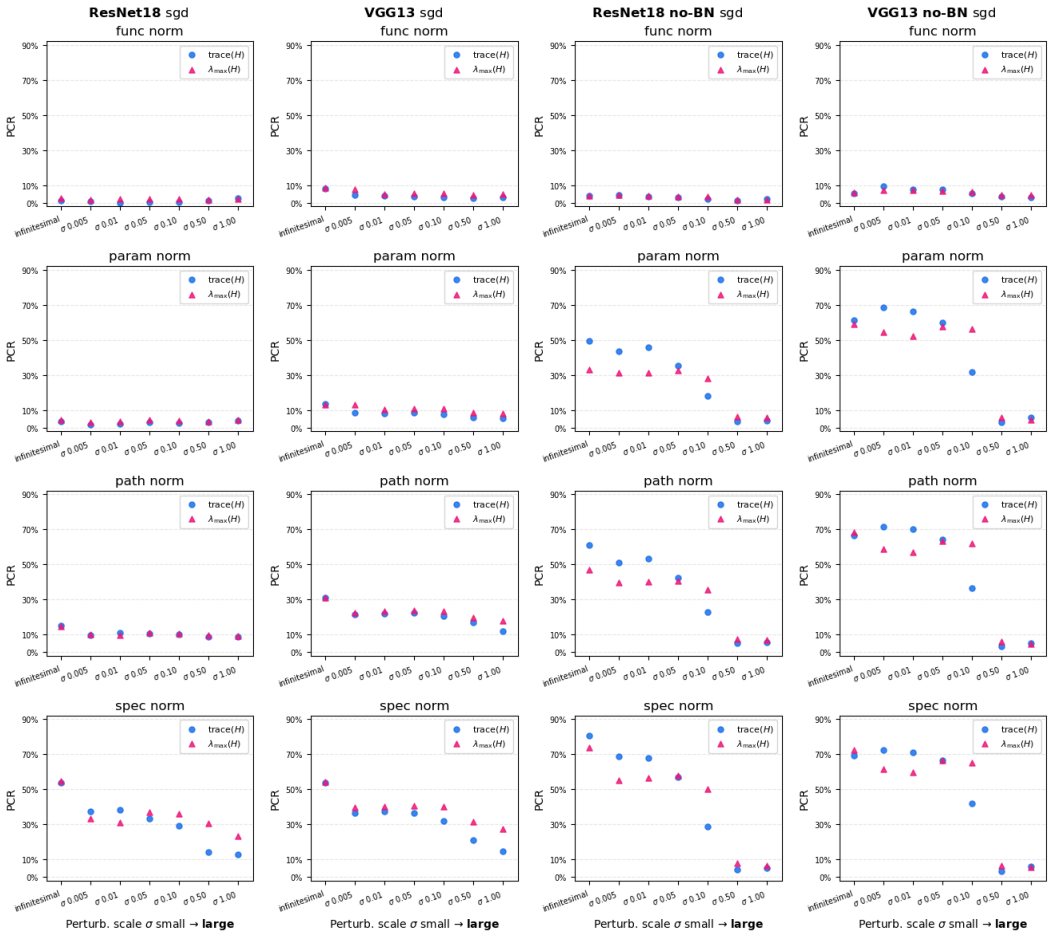
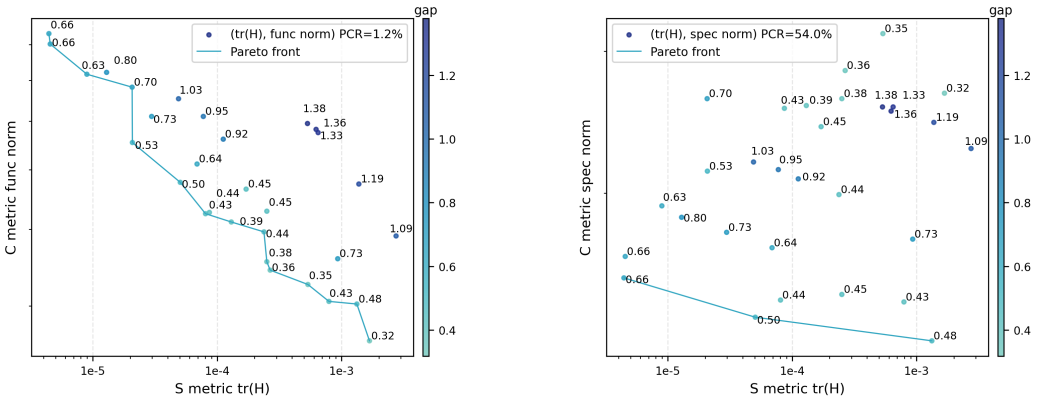


Figure 3: Full version of Figure 2. How PCR varies as the sharpness perturbation radius increases from an infinitesimal scale to small and then large finite scale. Each column corresponds to an architecture. Within each subplot, with the complexity metric fixed, sharpness defined over a *larger* neighborhood yields lower PCR, indicating stronger explanatory power and supporting Observation 2. The row-wise comparisons of subplots support Observations 4, 5, and 6.

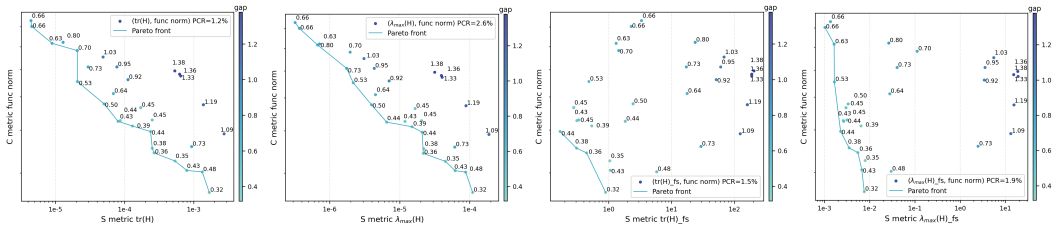
D.4 VISUALIZED PCRS



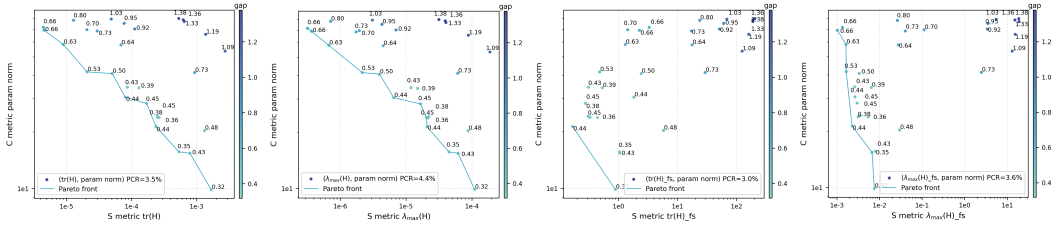
(a) (tr(H), func norm), PCR$=1.2\%$.

(b) (tr(H), spec norm), PCR$=54.0\%$.

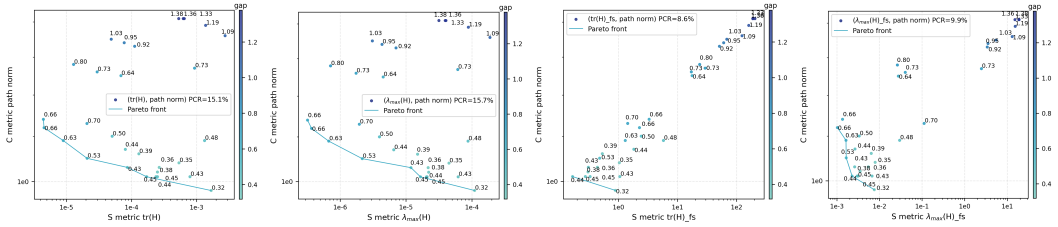
Figure 4: Two representative PCR visualizations in the sgd-trained ResNet18 setting, corresponding to the two PCR values listed in the first vertical block in Table 4. Lighter colors indicate smaller generalization gaps. Under the expected monotonic relationship, the lower-left point in each candidate pair should have a smaller generalization gap and therefore a lighter color, than the upper-right point. The left, metric pair (tr(H), func norm): few violations of the expected ordering, achieving low PCR 1.2%. The right, metric pair (tr(H), spec norm): severe violations of the expected ordering, achieving high PCR 54.0%.



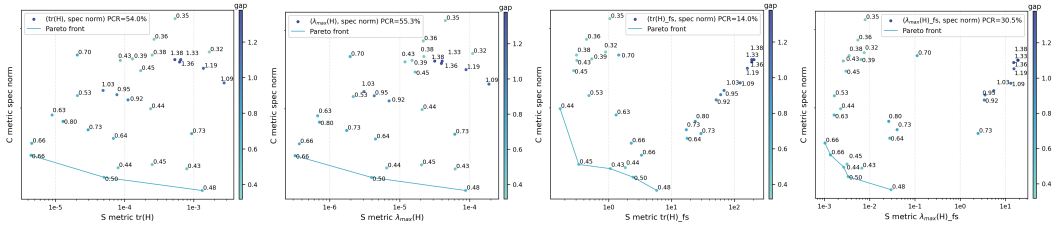
Func norm paired with four sharpness metrics.



Param norm paired with four sharpness metrics.



Path norm paired with four sharpness metrics.



Spec norm paired with four sharpness metrics.

Figure 5: Visualized PCRs for the sgd-trained ResNet18 setting, corresponding to the sixteen PCR values reported in the first vertical block in Table 4. Lighter colors indicate smaller generalization gaps. Under the expected monotonic relationship, the lower-left point in each candidate pair should have a smaller generalization gap and therefore a lighter color, than the upper-right point. If this expected ordering is never violated, the PCR is 0%; if it is frequently violated, the PCR becomes high.