

Article

Not peer-reviewed version

Are Flat Minima an Illusion?

Michael Timothy Bennett *

Posted Date: 25 March 2026

doi: 10.20944/preprints202603.1737.v2

Keywords: flat minima; sharpness; generalisation; reparameterisation invariance; weakness; PACBayes; large-batch training; model selection; ReLU networks; loss landscape



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Are Flat Minima an Illusion?

Michael Timothy Bennett

School of Computing, The Australian National University; michael.bennett@anu.edu.au

Abstract

Neural networks that land in flat regions of the loss landscape tend to generalise better than those in sharp regions. Sharpness-Aware Minimisation exploits this to improve generalisation. But function-preserving reparameterisation can inflate the Hessian of any minimum by two orders of magnitude without changing a single prediction. If the geometry of weight space can be manufactured from nothing, it cannot be the cause of anything. In other words, flat is simple and simplicity depends on encoding. Here I show that the actual driver is weakness, the volume of completions compatible with the learned function in the learner's embodied language. Weakness is reparameterisation-invariant because it is defined over what the network *does*, not how it is parameterised. I prove weakness is minimax-optimal under exchangeable demands, and that PAC-Bayes bounds work because they correlate with it. On MNIST, the large-batch generalisation advantage *vanishes* as training data grows, from +1.6% at $n = 2,000$ to +0.02% at $n = 60,000$. A quantity whose predictive power depends on how much data you have is not a cause but a confounder. I run head-to-heads on 100 networks with identical architecture and training. For MNIST weakness predicts generalisation ($\rho = +0.374$, $p = 0.00012$), sharpness anticorrelates ($\rho = -0.226$) and simplicity predicts nothing ($p = 0.848$). For Fashion-MNIST ($\rho = +0.384$, $p = 8.15 \times 10^{-5}$), though simplicity is at least somewhat predictive there. Simplicity is dataset dependent, whereas weakness is invariant. Flat minima were never the answer.

Keywords: flat minima; sharpness; generalisation; reparameterisation invariance; weakness; PACBayes; large-batch training; model selection; ReLU networks; loss landscape

1. Introduction

Flat minima generalise better than sharp ones. At least, that is the orthodox position. Hochreiter and Schmidhuber [1] argued it in 1997, Keskar et al. [2] demonstrated it empirically in 2017. Sharpness-Aware Minimisation [3] explicitly seeks flat minima and improves generalisation.

Here the correlation is not in dispute, but the causation is. Dinh et al. [4] gave reasons to doubt it. They constructed a reparameterisation that transforms sharp minima into a flat without changing the function of the network. The Hessian eigenvalues and loss landscape geometry change, while the function and predictions do not.

Several authors have proposed parameterisation-invariant sharpness measures. Tsuzuku et al. [5] normalised by parameter scale. Petzka et al. [6] used a Fisher information metric. Kwon et al. [7] introduced adaptive sharpness. They still do not answer the question "what property of the learned function makes it generalise?"

This paper offers three contributions.

1. **The vanishing advantage.** The generalisation advantage of large-batch over small-batch training vanishes as a function of training set size. On MNIST with a 3-layer ReLU MLP, large-batch networks generalise better at 500, 2,000, 6,000, and 12,000 training points. At 24,000 the gap is small (+0.08 pp, $p = 0.002$). At 60,000 it is negligible (+0.02 pp). A quantity that correlates at some scales and not others is not a cause. It is a confound.
2. **Reparameterisation invariance.** I prove that weakness [8,9] is reparameterisation-invariant by construction. I confirm non-invariance of the Hessian trace on trained networks, with changes of up to $99\times$ while generalisation stays exactly constant.

3. **Weakness beats sharpness.** I construct two formal vocabularies for frozen-partition ReLU networks. The region-class vocabulary (Appendix H) is an approximation that treats each activation region independently. The feature-classifier vocabulary (Appendix I) respects the shared-weight constraint exactly and reduces each extension check to a linear program. Bridging the lack of a direct neural-network analogue of formal weakness was identified as an open problem in Bennett [10]; the present work provides a concrete partial operationalisation via linear feasibility and pair-proxy measures. I measure weakness via linear feasibility on 100 networks with the same architecture, data, and training. Weakness outperforms sharpness at model selection ($\rho = +0.374$, $p = 0.00012$ vs $\rho = -0.226$, $p = 0.024$). Simplicity does not predict at all ($p = 0.848$). The result replicates on Fashion-MNIST at nearly identical strength ($\rho \approx +0.38$, $p \approx 10^{-4}$).

Buffer size (k_{free}) is a significant but modest predictor ($\rho = +0.117$, $p = 0.043$). The region-class vocabulary is an approximation. The feature-classifier vocabulary (Appendix I) solves the shared-weight problem and computes single-point extensions exactly, but the pair proxy is a sum of marginals, not the full extension count.

The structure is as follows. Section 2 reviews the definitions from Stack Theory. Section 3 extends weakness to continuous domains. Section 4 presents the reparameterisation invariance proof, the PAC-Bayes connection, and the confounding diagnosis. Section 5 presents the experiments. Section 6 locates continuous weakness in the landscape of generalisation theory. Section 7 discusses implications, limitations, and future work. Appendix H constructs the region-class vocabulary. Appendix I constructs the feature-classifier vocabulary, which respects the shared-weight constraint and reduces each extension check to a linear program.

The proofs are self contained, but the definitions used here drawn from the broader Stack Theory literature [11]. The finite-case predecessors of these continuous results were proved in [9].

2. Background

Definition 1 (Environment and programs). *An environment is a nonempty set Φ of mutually exclusive states. A program is any subset $p \subseteq \Phi$. Write $\mathcal{P} = 2^\Phi$ for the set of all programs. A vocabulary is any set of programs $\mathfrak{v} \subseteq \mathcal{P}$.*

Think of Φ as the set of all possible configurations of the world. Each state is one complete configuration. A program is a constraint that holds in some configurations and not others. A vocabulary is the set of constraints a system can express. For a neural network, the vocabulary is the set of input-output behaviours the architecture can realise.

Definition 2 (Embodied language and statements). *A vocabulary $\mathfrak{v}$ induces an embodied language*

$$L_{\mathfrak{v}} = \left\{ l \subseteq \mathfrak{v} \mid \bigcap_{p \in l} p \neq \emptyset \right\}.$$

Elements $l \in L_{\mathfrak{v}}$ are called statements. The truth set of l is $T(l) = \bigcap_{p \in l} p$.

A statement is a consistent conjunction of programs corresponding to a subset of possible physical states. For example, raising one's arm is a statement. It is a constraint on spatial position, musculature and so on. The embodied language $L_{\mathfrak{v}}$ is *not* the full powerset $2^{\mathfrak{v}}$. Most combinations of programs are mutually exclusive. You cannot raise and lower your arm at the same time. The satisfiability constraint $\bigcap_{p \in l} p \neq \emptyset$ is what makes the structure interesting.

Definition 3 (Extensions and weakness). *A completion of $x \in L_{\mathfrak{v}}$ is any $y \in L_{\mathfrak{v}}$ with $x \subseteq y$. The extension of x is*

$$\text{Ext}(x) = \{y \in L_{\mathfrak{v}} \mid x \subseteq y\}.$$

For a set of statements $X \subseteq L_v$, write $\text{Ext}(X) = \bigcup_{x \in X} \text{Ext}(x)$. The weakness of x is $w(x) = |\text{Ext}(x)|$.

Weakness measures how non-committal a statement is. The more commitments you can still make, the weaker the statement.

Definition 4 (Tasks, correctness, and learning). A v -task is a pair $\alpha = \langle I_\alpha, O_\alpha \rangle$ where $I_\alpha \subseteq L_v$ and $O_\alpha \subseteq \text{Ext}(I_\alpha)$. A policy $\pi \in L_v$ is correct for α if $\text{Ext}(I_\alpha) \cap \text{Ext}(\pi) = O_\alpha$. The set of correct policies is Π_α . Task α is a child of ω , written $\alpha \sqsubset \omega$, if $I_\alpha \subsetneq I_\omega$ and $O_\alpha \subseteq O_\omega$.

A task specifies inputs and correct outputs. A policy constrains how the system completes inputs. Correctness means the policy produces exactly the right outputs on the given inputs. A child task is a task with fewer examples. Learning means generalising from a child task to its parent.

The foundational result is that under an exchangeable prior over parent tasks, weakness maximisation is both necessary and sufficient for optimal generalisation [8,9]. Among all correct policies for the child, the ones most likely to also be correct for the unknown parent are the weakest. Minimising description length is neither necessary nor sufficient [12].

3. Extending Weakness to Continuous Domains

When the vocabulary is finite, weakness is a count. You enumerate completions and take the cardinality. When the vocabulary is infinite, every uncountable extension has the same cardinality, so counting breaks. This section replaces counting with measuring.

Definition 5 (Measurable vocabulary). A measurable vocabulary is a triple $(v, \mathcal{A}_L, \mu)$ where $v \subseteq \mathcal{P}$ is a vocabulary (possibly infinite), $\mathcal{A}_L$ is a σ -algebra on L_v , and μ is a σ -finite measure on $(L_v, \mathcal{A}_L)$. I require that $\text{Ext}(l)$ is $\mathcal{A}_L$ -measurable for every $l \in L_v$. Since $\text{Ext}(X) = \bigcup_{x \in X} \text{Ext}(x)$ for any $X \subseteq L_v$, and a countable union of measurable sets is measurable, $\text{Ext}(I_\alpha)$ is measurable whenever I_α is countable. All tasks in this paper have finite input sets.

The measure μ does for infinite vocabularies what counting measure does for finite ones. It assigns a size to sets of statements so that extensions can be compared. When v is finite, setting μ to counting measure recovers every existing result.

Definition 6 (Continuous weakness). The μ -weakness of $l \in L_v$ is $w_\mu(l) = \mu(\text{Ext}(l))$.

Definition 7 (Continuous extension model). Fix a measurable vocabulary $(v, \mathcal{A}_L, \mu)$ and a task α with output region O_α . Define the unseen region $U = L_v \setminus \text{Ext}(I_\alpha)$, where $0 < \mu(U) < \infty$ and $\mu(O_\alpha) < \infty$. Assume $(U, \mathcal{A}_L|_U, \mu|_U)$ is an atomless standard measure space (Appendix A). A continuous extension model is a probability space $(\Omega, \mathcal{F}, P)$ with a random set $S: \Omega \rightarrow 2^U$ such that $\{S \subseteq B\}$ and $\{S \subseteq B, S \cap A \neq \emptyset\}$ are $\mathcal{F}$ -measurable for every $A, B \in \mathcal{A}_L|_U$. The buffer of a correct policy π is $B_\pi = \text{Ext}(\pi) \cap U$. The P -weakness is $w_P(\pi) = P(S \subseteq B_\pi)$.

The parent task will additionally demand some set S of outputs. The policy survives if and only if every demand falls inside its buffer.

Definition 8 (μ -exchangeability and nondegeneracy). A continuous extension model is μ -exchangeable if for every measure-preserving bijection $\sigma: (U, \mu) \rightarrow (U, \mu)$ and every measurable $B \subseteq U$, $P(S \subseteq B) = P(S \subseteq \sigma(B))$. It is nondegenerate if for every measurable $A \subseteq U$ with $\mu(A) > 0$, we have $P(\emptyset \neq S \subseteq A) > 0$. (This event is measurable because $\{\emptyset \neq S \subseteq A\} = \{S \subseteq A\} \setminus \{S \subseteq \emptyset\}$.)

Write $G(\pi, P) = w_P(\pi) = P(S \subseteq B_\pi)$ for the generalisation probability of policy π under extension model P .

Lemma 1 (Buffer measure determines μ -weakness for correct policies). *For any correct $\pi \in \Pi_\alpha$, $w_\mu(\pi) = \mu(O_\alpha) + \mu(B_\pi)$, where $O_\alpha = \text{Ext}(I_\alpha) \cap \text{Ext}(\pi)$ is the output region (fixed for all $\pi \in \Pi_\alpha$ by the correctness condition) and $B_\pi = \text{Ext}(\pi) \cap U$. μ -weakness ranking and buffer-measure ranking agree among correct policies.*

Proof. By the correctness condition, $\text{Ext}(\pi) = O_\alpha \sqcup B_\pi$ where $O_\alpha \subseteq \text{Ext}(I_\alpha)$ and $B_\pi \subseteq U = L_v \setminus \text{Ext}(I_\alpha)$. These sets are disjoint and both $\mathcal{A}_L$ -measurable (by the measurability requirement on extensions in Definition 5). So $w_\mu(\pi) = \mu(\text{Ext}(\pi)) = \mu(O_\alpha) + \mu(B_\pi)$. Since $\mu(O_\alpha)$ is the same for all $\pi \in \Pi_\alpha$, ranking by $w_\mu(\pi)$ is equivalent to ranking by $\mu(B_\pi)$. $\square$

The continuous theory produces four main theorems. Full proofs are in Appendices C–F.

Theorem 1 (Sufficiency). *If P is μ -exchangeable, then $w_P(\pi) = f(\mu(B_\pi))$ for some nondecreasing f . Among correct policies, μ -weakness determines generalisation probability.*

Theorem 2 (Strict necessity). *If P is μ -exchangeable and nondegenerate, and $\mu(B_{\pi_1}) > \mu(B_{\pi_2})$, then $w_P(\pi_1) > w_P(\pi_2)$.*

Theorem 3 (Unified optimality). *Let $\mathcal{E}$ be the class of all μ -exchangeable models.*

1. Sufficiency. $\mu(B_{\pi_1}) \geq \mu(B_{\pi_2}) \implies G(\pi_1, P) \geq G(\pi_2, P)$ for all $P \in \mathcal{E}$.
2. Strict necessity. If P is nondegenerate, $\mu(B_{\pi_1}) > \mu(B_{\pi_2}) \implies G(\pi_1, P) > G(\pi_2, P)$.
3. Uniqueness. All optimal policies have the same buffer measure $\mu(B_\pi)$, hence the same μ -weakness.

Theorem 4 (Minimax optimality). *If a μ -weakest correct policy π^* exists,*

$$\max_{\pi \in \Pi_\alpha} \inf_{P \in \mathcal{E}} G(\pi, P) = \inf_{P \in \mathcal{E}} \max_{\pi \in \Pi_\alpha} G(\pi, P),$$

both achieved by π^ .*

This result is a corollary of sufficiency. Because π^* maximises $G(\pi, P)$ for every $P \in \mathcal{E}$ simultaneously, it trivially achieves the max-min and the minimax equality holds. The learner has a dominant strategy. No adversary can make weakness maximisation suboptimal.

4. The Flat Minima Resolution

The continuous theory resolves the flat minima puzzle. The resolution has three parts.

4.1. Reparameterisation invariance of weakness

Definition 9 (Policy map and function-preserving reparameterisation). *Fix a measurable vocabulary defined over input-output behaviour. Let $\Theta \subseteq \mathbb{R}^d$ be a parameter space and let $F(\theta)$ denote the function computed by the network with parameters θ . A program $p \in \mathfrak{v}$ is satisfied by $F(\theta)$ if the input-output behaviour of $F(\theta)$ lies in p (viewed as a subset of Φ). The policy map is*

$$\Pi_F(\theta) = \{p \in \mathfrak{v} \mid F(\theta) \text{ satisfies } p\}.$$

A function-preserving reparameterisation is a diffeomorphism $\phi: \Theta \rightarrow \Theta$ with $F(\phi(\theta)) = F(\theta)$ for all θ .

Theorem 5 (Reparameterisation invariance of μ -weakness). *If ϕ is function-preserving, then $\Pi_F(\phi(\theta)) = \Pi_F(\theta)$, hence $w_\mu(\Pi_F(\phi(\theta))) = w_\mu(\Pi_F(\theta))$.*

Proof. $F(\phi(\theta)) = F(\theta)$ by assumption. The policy map depends only on which programs $F(\theta)$ satisfies. Same function, same programs satisfied, same policy, same extension, same weakness. $\square$

A three sentence proof. The definitions do most of the work. The vocabulary lives in function space. The reparameterisation does not move you in function space, so weakness invariant.

4.2. PAC-Bayes Bounds Track Weakness

Theorem 6 (PAC-Bayes ordering agrees with μ -weakness). *Fix a measurable vocabulary with $\mu(L_v) < \infty$. Set $P_0 = \mu / \mu(L_v)$ (the normalised prior) and $Q_\pi = \mu|_{B_\pi} / \mu(B_\pi)$ (uniform posterior on the buffer). Then for correct π with $\mu(B_\pi) > 0$,*

$$\text{KL}(Q_\pi \| P_0) = \log \frac{\mu(L_v)}{\mu(B_\pi)},$$

strictly decreasing in $\mu(B_\pi)$.

Proof. The Radon-Nikodym derivative is $dQ_\pi / dP_0 = \mu(L_v) / \mu(B_\pi) \cdot \mathbf{1}_{B_\pi}$. Computing the KL-divergence,

$$\text{KL}(Q_\pi \| P_0) = \int \log \frac{dQ_\pi}{dP_0} dQ_\pi = \log \frac{\mu(L_v)}{\mu(B_\pi)}.$$

Since $\mu(O_\alpha)$ is fixed for all $\pi \in \Pi_\alpha$ and $\mu(\text{Ext}(\pi)) = \mu(O_\alpha) + \mu(B_\pi)$, the KL is strictly decreasing in $\mu(B_\pi)$, hence strictly decreasing in μ -weakness $w_\mu(\pi) = \mu(\text{Ext}(\pi))$. $\square$

Since the standard PAC-Bayes bound is increasing in $\text{KL}(Q_\pi \| P_0)$ for fixed empirical risk and sample size, and correct policies have zero empirical risk by definition, the μ -weakest correct policy minimises the PAC-Bayes bound. PAC-Bayes bounds work because they track weakness. The KL-divergence is a proxy for a proxy. The fundamental quantity is the buffer measure.

4.3. Diagnosis of Confounding

The preceding results assemble the diagnosis.

1. Weakness is reparameterisation-invariant (Theorem 5).
2. Weakness determines generalisation probability under exchangeable demands (Theorems 1–4).
3. Sharpness is not reparameterisation-invariant [4].
4. Therefore sharpness is not the driver.

5. Experiments

5.1. Setup

The cross-regime experiments (Sections 5.2–5.4) use a 3-layer ReLU MLP (784→256→128→10) trained on subsets of MNIST with no regularisation. The model-selection experiments (Section 5.6 and Section 5.7) use smaller architectures as specified in each section. I train two regimes per data scale. The *small-batch* regime uses batch size 64 with learning rate 0.01–0.05. The *large-batch* regime uses batch size equal to the training set size at 500 and 2,000, and batch size 4,096 at 6,000 and above, with learning rate 0.2–0.5. All networks are trained to >99.5% training accuracy. At most scales, all networks reach exactly 100%. Exceptions are one network at 2,000 (99.95%), the 24,000 large-batch regime (99.98–100%), and 4 networks at 60,000 (99.998%). 25 networks per regime, 50 per data scale.

For the reparameterisation test, I use $T_\beta(\theta) = (\beta W_1, \beta b_1, W_2 / \beta, b_2, W_3, b_3)$ between layers 1 and 2, and $T_\gamma(\theta) = (W_1, b_1, \gamma W_2, \gamma b_2, W_3 / \gamma, b_3)$ between layers 2 and 3. By the positive homogeneity of ReLU, both preserve the function computed by the network. The 3-layer architecture provides two independent reparameterisation axes. For each network I measure:

1. **Hessian trace.** Estimated via Hutchinson’s method with 50 Rademacher vectors and autodifferentiation Hessian-vector products.
2. **Layer-1 activation patterns.** The number of distinct binary activation patterns $\mathbf{1}[W_1 x + b_1 > 0]$ across all unseen points.
3. **Layer-2 activation patterns.** The number of distinct binary activation patterns $\mathbf{1}[W_2 h_1 + b_2 > 0]$ across all unseen points.
4. **Ensemble agreement.** For each network, among the unseen points it misclassifies, the fraction correctly classified by other networks in the same experiment.

5.2. The Vanishing Advantage

Table 1 presents test accuracy by regime across six training set sizes. At 500, 2,000, 6,000, and 12,000 training points, the large-batch regime generalises better. At 24,000 the gap is small but statistically significant (+0.08 pp, Welch $p = 0.002$). At 60,000 it is negligible (+0.02 pp). The large-batch advantage peaks at 2,000 training points and shrinks thereafter.

Table 1. Test accuracy by training regime across data scales. All rows use the 3-layer architecture. †Large-batch networks that did not reach exactly 100% training accuracy (one at 2,000 at 99.95%, 23 at 24,000 (>99.98%), 4 at 60,000 at 99.998%). All other networks reached exactly 100%.

n	Small-batch	Large-batch	Δ	Hessian (small / large)
500	85.3%	86.5%	+1.2	52.5 / 20.6
2,000	90.4%	92.0%	+1.6 [†]	59.5 / 23.1
6,000	94.0%	94.9%	+1.0	61.9 / 25.9
12,000	96.0%	96.5%	+0.5	44.8 / 23.9
24,000	97.0%	97.1%	+0.08 [†]	39.1 / 38.2
60,000	98.0%	98.0%	+0.02 [†]	30.1 / 26.0

At every data scale, the large-batch regime has *lower* Hessian trace (Table 1). The standard association between large batch size and sharp minima [2] is reversed at every scale tested. The Hessian tracks the generalisation gap but does not explain it.

5.3. Reparameterisation Invariance on Trained Networks

I apply T_β and T_γ to trained networks at three data scales (500, 2,000, 6,000) and measure all four quantities. The Hessian trace changes by up to $99\times$ under T_β and $54\times$ under T_γ , while test accuracy, L1 and L2 activation patterns, and ensemble agreement are exactly invariant to the precision of floating-point arithmetic. This holds for all 12 networks tested across all three data scales. The two independent reparameterisation axes (T_β between layers 1–2, T_γ between layers 2–3) demonstrate that the argument generalises to arbitrary depth. Full results are in Appendix G.

5.4. Correlations with Generalisation

Table 2 presents Spearman rank correlations between each measure and test accuracy across all 50 networks at each data scale.

Table 2. Spearman correlations (ρ) with test accuracy. L1 and L2 are layer-1 and layer-2 activation pattern counts on unseen data. EA is ensemble agreement. Bold indicates $|\rho| > 0.7$. The unseen set comprises leftover training data plus the test set at 500–12,000, leftover training data only at 24,000, and the test set only at 60,000.

n	Hessian	L1	L2	EA
500	−0.75	+0.11	+0.76	−0.85
2,000	−0.82	+0.08	+0.73	−0.88
6,000	−0.78	+0.40	+0.78	−0.86
12,000	−0.84	+0.22	+0.77	−0.90
24,000	+0.01	+0.06	+0.39	−0.40
60,000	+0.05	—	+0.05	−0.81

Hessian trace and L2 activation pattern count both predict generalisation at $|\rho| \approx 0.75$ across 500–12,000. At 24,000, the Hessian collapses ($\rho = +0.01$) while L2 remains significant ($\rho = +0.39$, $p = 0.005$). L1 saturates. EA is confounded by error count.

The key comparison is between the Hessian and L2 columns. Both predict. But the Hessian is *not* reparameterisation-invariant. L2 is exactly invariant. Two quantities predict equally well. One is an artefact of the parameterisation. The other is a property of the function.

5.5. Connecting L2 Diversity to Weakness

L2 activation pattern diversity is not, on its face, a weakness measure. More patterns means more commitments at layer 2, not fewer. The connection to weakness runs through the training data and the formal construction in Appendix H.

Freeze layers 1 and 2. This fixes the L2 partition of input space. Within each L2 region, $h(x)$ is affine in x , so the output $W_3h(x) + b_3$ is also affine in x . Different inputs in the same region can in general receive different classifications. The region-class vocabulary in Appendix H approximates this by treating each region as assigning a single class. This approximation is exact when each region contains at most one data point. In the cross-regime experiments at smaller scales, L2 regions outnumber training points but not unseen points. At 60,000, L2 regions (approximately 9,500) are far fewer than training points. The approximation is therefore inexact in all experiments reported here. Define a *region-class vocabulary* (Definition A1) where programs are of the form “region r is classified as class c .” The embodied language is the set of partial functions from regions to classes (Proposition A3).

An L2 region is *free* if it contains unseen data but no training data. In these regions, the network has committed to a classification, but the child task provides no constraint on what that classification should be. Theorem A2 proves that the weakness of the training-label policy is $w(\pi) = 11^k$ where k is the number of free regions. Under the region-class vocabulary (an approximation), every partial classification of free regions extends the training-label policy without contradiction. In the actual network, the shared weight matrix W_3 couples the classifications across regions, so the choices are not independent. See Section H.6 for details.

At 6,000 training points, the small-batch networks have $\sim 54,000$ L2 regions and the large-batch networks have $\sim 63,000$. With 6,000 points spread across this many regions, the vast majority of regions contain zero training points. The large-batch networks have $\sim 9,000$ more L2 regions. Under the region-class approximation, more L2 regions means more free regions and therefore higher weakness (Corollary A1). The cross-regime L2 diversity correlation ($\rho = 0.73\text{--}0.78$, Table 2) is consistent with this prediction, though L2 pattern count and free-region count are not identical quantities.

There is a caveat (Section H.6). The region-class vocabulary treats each free region independently. In reality, the shared weight matrix W_3 creates correlations between regions. The vocabulary overstates the freedom of the network. Whether the ranking by free regions is preserved under the shared-weight constraint is an open question that the experiments test empirically.

5.6. Model Selection by Weakness

The cross-regime results in Tables 1–2 compare networks from different training regimes. A stronger test is model selection *within* a single regime. I train 300 networks with the same architecture ($784 \rightarrow 64 \rightarrow 16 \rightarrow 10$), the same 250 training points, the same batch size, the same learning rate, and different random seeds. All memorise the training data. I measure six quantities per network. Three are not reparameterisation-invariant (Hessian trace, weight L2 norm, weight L1 norm). Three are invariant (L2 unseen patterns, free parameters (defined in Table 3 caption), unseen-only regions = k_{free}).

Table 3. Model selection within a single regime ($n = 300$ networks, same config, different seeds). Spearman ρ with test accuracy. Bold indicates statistically significant. The two significant predictors are both reparameterisation-invariant. No sharpness or simplicity measure predicts. *Free parameters = $\sum_r \max(0, D_2 \cdot 10 + 10 - n_r)$ where n_r is the number of training points in region r .

Measure	ρ	p	Invariant?
Hessian trace	−0.079	0.174	No
Weight L2 norm	−0.061	0.293	No
Weight L1 norm	+0.011	0.845	No
L2 unseen patterns	+ 0.113	0.050	Yes
Free parameters*	+0.067	0.249	Yes
Unseen-only regions (k_{free})	+ 0.117	0.043	Yes

Unseen-only regions (k_{free}) is statistically significant ($\rho = +0.117$, $p = 0.043$). This is the quantity for which Theorem A2 proves the weakness is 11^k under the region-class vocabulary (an approximation to the actual network; see Section H.5). The sufficiency theorem predicts that weaker policies generalise better within a fixed vocabulary. The data confirms this prediction empirically, though comparing across networks involves different vocabularies (see Appendix H for discussion). L2 unseen patterns is also significant ($p = 0.050$). The effect is small ($\rho \approx 0.12$), but the direction is correct and backed by a theorem. Sharpness or simplicity fail to predict generalisation here.

5.7. Pair Proxy

A separate experiment tests the pair proxy directly. I train 100 networks ($784 \rightarrow 64 \rightarrow 8 \rightarrow 10$, 250 training points, same batch size, different seeds). For each network, I freeze layers 1–2 and solve a linear feasibility problem for each of 100 unseen inputs $\times$ 10 classes. The linear program asks whether there exists a setting of layer-3 weights that correctly classifies all training data AND assigns the unseen input to the candidate class. The count of feasible pairs is the pair proxy.

Table 4. Pair proxy experiment ($n = 100$ networks, $784 \rightarrow 64 \rightarrow 8 \rightarrow 10$, 250 training points). Weakness outperforms sharpness. Simplicity does not predict.

Measure	ρ	p	Invariant?
Hessian trace	−0.226	0.024	No
Weight norm	+0.019	0.848	No
Pair proxy	+ 0.374	1.2×10^{-4}	Yes

The pair proxy is highly significant ($\rho = +0.374$, $p = 0.00012$). It outperforms sharpness ($\rho = -0.226$, $p = 0.024$). Simplicity does not predict at all ($p = 0.848$) for MNIST, and is inconsistent across datasets when it does work. The pair proxy is reparameterisation-invariant while the Hessian trace is not. This is a head-to-head comparison on the same 100 networks. Same architecture, same data, same training, different random seeds and as you might expect the answer is weakness.

5.8. Replication on Fashion-MNIST

I replicate the pair proxy experiment on Fashion-MNIST [13] with the same architecture ($784 \rightarrow 64 \rightarrow 8 \rightarrow 10$), same number of training points (250), same number of networks (100), and same number of unseen probe points (100). The *only* difference is the dataset. Fashion-MNIST uses 28×28 grayscale images of clothing items instead of handwritten digits. It is harder than MNIST (mean test accuracy 71.8% vs $\sim 80\%$ on MNIST with this architecture and training set size).

Table 5. Fashion-MNIST pair proxy experiment ($n = 100$ networks, $784 \rightarrow 64 \rightarrow 8 \rightarrow 10$, 250 training points). Same setup as Table 4.

Measure	ρ	p	Invariant?
Hessian trace	−0.241	0.016	No
Weight norm	−0.226	0.024	No
Pair proxy	+0.384	8.15×10^{-5}	Yes

The pair proxy predicts generalisation at $\rho = +0.384$ ($p = 8.15 \times 10^{-5}$), closely matching the MNIST result ($\rho = +0.374$, $p = 0.00012$). Same architecture, same optimisation, same evaluation protocol, different data, same answer.

Weight norm is also significant on Fashion-MNIST ($\rho = -0.226$, $p = 0.024$), but the effect is modest and dataset-dependent. On MNIST it does not predict at all ($p = 0.848$). By contrast, the pair proxy is consistent across both datasets and is reparameterisation-invariant by construction. This is consistent with the theoretical prediction that simplicity is not the driver [12].

6. Where μ -Weakness Sits

Generalisation theory has produced many complexity measures. Most of them answer the wrong question, or the right question at the wrong level.

VC dimension [14] measures the expressiveness of a hypothesis *class*. Weakness measures the committedness of a single *hypothesis*. VC dimension cannot distinguish two hypotheses within the same class.

Algorithmic stability [15] measures sensitivity of the learning *algorithm* to perturbation of a single training point. Weakness is a property of the output, not the algorithm that produced it.

PAC-Bayes bounds [16] operate at the right level. Theorem 6 shows the KL term is a monotone function of weakness among correct policies. The difference is that weakness is defined in terms of the embodied language, which makes reparameterisation invariance obvious.

Minimum description length [17] is a property of form, not function. Description length is encoding-dependent and can be gamed. Weakness cannot. If a persisting object happens to be simple, that is incidental unless it is also weak.

Sharpness [1,2] is a geometric property of weight space. It is not reparameterisation-invariant. The vanishing advantage (Section 5.2) shows it is not even a consistent correlate of generalisation.

AIXI [18] selects policies by minimising description length. Weakness maximisation is Bayes optimal under exchangeable task growth [9,12]. AIXI's Occam rule is not.

Maximum entropy. Under a uniform prior over parent tasks, weakness maximisation is optimal (Proposition A1). A uniform prior over tasks is a maximum-entropy prior over which parent will appear. So weakness maximisation is the optimal response to a maximum-entropy assumption about tasks. But tasks and outputs are different things. A uniform prior over tasks implies a uniform distribution over outputs only if every combination of output constraints is satisfiable, which requires $L_v = 2^v$. This is the degenerate case where weakness collapses to a cardinality proxy and maximum entropy over outputs coincides with weakness maximisation [12]. In the non-degenerate cases ($L_v \neq 2^v$), the satisfiability constraint filters the uniform prior over tasks into a non-uniform distribution over achievable outputs. Hence maximum entropy and weakness differ whenever the vocabulary has non-trivial satisfiability structure, which is almost always.

7. Discussion

Limitations. The pair proxy outperforms sharpness but explains a modest fraction of within-regime variance ($\rho \approx 0.37$ on both datasets). The region-class vocabulary is an approximation. The feature-classifier vocabulary (Appendix I) solves the shared-weight problem but the pair proxy sums single-point marginals, not the full extension. Isolated networks did not reach exactly 100% training accuracy (see Table 1 caption). The experiments use MLPs on MNIST and Fashion-MNIST. The continuous weakness theory requires a reference measure μ not present in the finite case.

Future work. Computing the full extension $|\text{Ext}(\pi)|$ in the feature-classifier vocabulary (not just single-point marginals) and extending the sufficiency theorem to account for extension quality (not just count) are the main open problems. This work partially addresses the operationalisation gap identified in Bennett [10], but a full characterisation of $|\text{Ext}(\pi)|$ for continuous function classes remains open. Extending the pair proxy to CNNs and transformers is the empirical priority.

Appendix A. Technical Assumption

Throughout the proofs below, fix a measurable vocabulary $(v, \mathcal{A}_L, \mu)$ with $0 < \mu(U) < \infty$. Assume $(U, \mathcal{A}_L|_U, \mu|_U)$ is an *atomless standard measure space*, meaning it is isomorphic (as a measure space) to a Lebesgue interval. This guarantees the existence of measure-preserving bijections between measurable subsets of equal measure, by Carathéodory's isomorphism theorem. The atomless requirement excludes pathological cases where atoms of equal measure but different demand probability would violate sufficiency. This assumption is satisfied by Lebesgue measure on $\mathbb{R}^d$, which covers all cases of

practical interest for neural networks. The finite case, where counting measure has atoms, is handled separately by the finite proofs in Appendix B.

Appendix B. Finite Sufficiency and Necessity (Prior Results)

These results are from Bennett [8] and Bennett [9]. I include the proofs for self-containment.

Proposition A1 (Finite sufficiency). *Let $\alpha \sqsubset \omega$ with the parent's additional demands drawn from the maximally uninformative extension model (uniform over $S \subseteq U$ where $U = L_{\omega} \setminus \text{Ext}(I_{\alpha})$), and let $\pi \in \Pi_{\alpha}$. Then $P(\pi \in \Pi_{\omega}) = 2^{|B_{\pi}|} / 2^{|U|}$ where $B_{\pi} = \text{Ext}(\pi) \cap U$.*

Proof. The parent adds demands by selecting $S \subseteq U$ uniformly from 2^U . The policy π survives if and only if $S \subseteq B_{\pi}$. There are $2^{|U|}$ equally likely subsets of U and $2^{|B_{\pi}|}$ of them lie inside B_{π} . For correct policies, $|B_{\pi}| = |\text{Ext}(\pi)| - |O_{\alpha}|$ and $|O_{\alpha}|$ is fixed, so maximising $|\text{Ext}(\pi)|$ is equivalent to maximising $|B_{\pi}|$ and therefore maximises the generalisation probability. $\square$

Proposition A2 (Finite necessity). *If $|\text{Ext}(\pi_1)| > |\text{Ext}(\pi_2)|$, then $P(\pi_1 \in \Pi_{\omega}) > P(\pi_2 \in \Pi_{\omega})$.*

Proof. Immediate from Proposition A1. $\square$

Theorem A1 (Finite exchangeable generalisation). *Under any exchangeable prior over parent tasks, $|\text{Ext}(\pi_1)| \geq |\text{Ext}(\pi_2)|$ implies $P(\pi_1 \in \Pi_{\omega}) \geq P(\pi_2 \in \Pi_{\omega})$. Under nondegeneracy, the inequality is strict when the extension sizes differ.*

Proof. Under exchangeability, the prior treats all completions symmetrically. A larger extension captures at least as many completions under any exchangeable measure. Nondegeneracy guarantees that the extra completions in the larger extension have positive probability. $\square$

Appendix C. Proof of Theorem 1 (Sufficiency Under μ -Exchangeable Demands)

Proof. Let $\mu(B_{\pi_1}) \leq \mu(B_{\pi_2})$.

Step 1. Choose a measurable $B'_2 \subseteq B_{\pi_2}$ with $\mu(B'_2) = \mu(B_{\pi_1})$ (exists by the atomless standard measure space assumption, Appendix A). Build a measure-preserving bijection $\sigma_1: B_{\pi_1} \rightarrow B'_2$ and a measure-preserving bijection $\sigma_2: U \setminus B_{\pi_1} \rightarrow U \setminus B'_2$ (both pairs have equal measure). Define $\sigma = \sigma_1$ on B_{π_1} and $\sigma = \sigma_2$ on $U \setminus B_{\pi_1}$. Then $\sigma: U \rightarrow U$ is a measure-preserving bijection with $\sigma(B_{\pi_1}) = B'_2 \subseteq B_{\pi_2}$.

Step 2. If $S \subseteq B_{\pi_1}$, then $\sigma(S) \subseteq \sigma(B_{\pi_1}) = B'_2 \subseteq B_{\pi_2}$.

Step 3. By μ -exchangeability, $P(S \subseteq B_{\pi_1}) = P(S \subseteq \sigma(B_{\pi_1})) = P(S \subseteq B'_2)$. Since $B'_2 \subseteq B_{\pi_2}$, $P(S \subseteq B'_2) \leq P(S \subseteq B_{\pi_2})$.

Step 4. Therefore $w_P(\pi_1) \leq w_P(\pi_2)$. Set $f(t) = w_P(\pi)$ for any π with $\mu(B_{\pi}) = t$. $\square$

Appendix D. Proof of Theorem 2 (Strict Necessity)

Proof. Step 1. Since $\mu(B_{\pi_1}) > \mu(B_{\pi_2})$, choose $B'_1 \subseteq B_{\pi_1}$ with $\mu(B'_1) = \mu(B_{\pi_2})$. Let $D = B_{\pi_1} \setminus B'_1$, so $\mu(D) > 0$. By the two-piece construction in Theorem 1, any two buffers of equal measure give equal generalisation probability. So $P(S \subseteq B_{\pi_2}) = P(S \subseteq B'_1)$.

Step 2. Since $B'_1 \subsetneq B_{\pi_1}$,

$$P(S \subseteq B_{\pi_1}) = P(S \subseteq B'_1) + P(S \subseteq B_{\pi_1}, S \cap D \neq \emptyset).$$

Step 3. The second term is at least $P(\emptyset \neq S \subseteq D)$. By nondegeneracy and $\mu(D) > 0$, this is strictly positive. Therefore $P(S \subseteq B_{\pi_1}) > P(S \subseteq B'_1) = P(S \subseteq B_{\pi_2})$, giving $w_P(\pi_1) > w_P(\pi_2)$. $\square$

Appendix E. Proof of Theorem 3 (Unified Optimality)

Proof. Part (1) is Theorem 1. Part (2) is Theorem 2. For Part (3), suppose $\pi_1, \pi_2 \in \Pi_\alpha$ both maximise $G(\pi, P)$ over Π_α for every nondegenerate $P \in \mathcal{E}$. If $\mu(B_{\pi_1}) \neq \mu(B_{\pi_2})$, say $\mu(B_{\pi_1}) > \mu(B_{\pi_2})$, then Part (2) gives $G(\pi_1, P) > G(\pi_2, P)$ for every nondegenerate P , so π_2 does not maximise $G(\cdot, P)$. Contradiction. So $\mu(B_{\pi_1}) = \mu(B_{\pi_2})$. $\square$

Appendix F. Proof of Theorem 4 (Minimax Optimality)

Proof. By Theorem 3(1), $G(\pi^*, P) \geq G(\pi, P)$ for all π and all $P \in \mathcal{E}$. So $\max_\pi G(\pi, P) = G(\pi^*, P)$ for every P , giving

$$\inf_{P \in \mathcal{E}} \max_\pi G(\pi, P) = \inf_{P \in \mathcal{E}} G(\pi^*, P).$$

Sufficiency (Theorem 3(1)) gives $G(\pi^*, P) \geq G(\pi, P)$ for all π and all $P \in \mathcal{E}$, so

$$\inf_{P \in \mathcal{E}} G(\pi^*, P) \geq \inf_{P \in \mathcal{E}} G(\pi, P) \quad \text{for all } \pi.$$

Therefore π^* achieves $\max_\pi \inf_P G(\pi, P)$. The minimax equality follows from the chain

$$\max_\pi \inf_P G(\pi, P) \geq \inf_P G(\pi^*, P) = \inf_P \max_\pi G(\pi, P) \geq \max_\pi \inf_P G(\pi, P).$$

$\square$

Appendix G. Reparameterisation Results

Table A1 shows detailed results for one small-batch network at 6,000 training points. In every case, test accuracy, L1, L2, and EA are exactly invariant under both T_β and T_γ . Only the Hessian trace changes.

Table A1. Reparameterisation invariance test. Net 0 (small-batch regime, 6,000 training points). Hessian trace changes by up to $99\times$. All other quantities are exactly invariant.

Reparam	Value	Test acc	Hessian	L1	L2	EA
T_β	1	0.9392	66.1	63,998	51,348	0.1909
T_β	2	0.9392	81.3	63,998	51,348	0.1909
T_β	5	0.9392	388.8	63,998	51,348	0.1909
T_β	10	0.9392	1,607.1	63,998	51,348	0.1909
T_β	20	0.9392	6,525.9	63,998	51,348	0.1909
T_γ	1	0.9392	62.9	63,998	51,348	0.1909
T_γ	5	0.9392	255.2	63,998	51,348	0.1909
T_γ	20	0.9392	3,378.9	63,998	51,348	0.1909

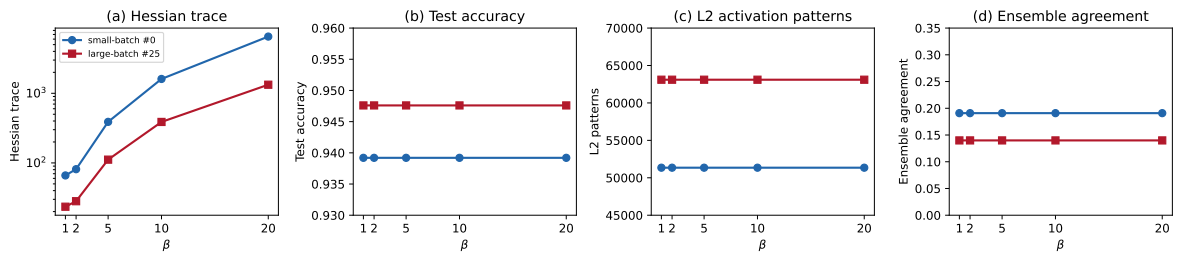


Figure A1. Reparameterisation invariance under T_β at 6,000 training points. (a) Hessian trace changes by two orders of magnitude. (b) Test accuracy is exactly invariant. (c) Layer-2 activation pattern count is exactly invariant. (d) Ensemble agreement is exactly invariant.

500 training points. Net 0 (small-batch, test=0.8536). Hessian at $\beta = 1$: 48.9. Hessian at $\beta = 20$: 4,751 ($97\times$). Test accuracy: 0.8536 at all β . Net 25 (large-batch, test=0.8619). Hessian at $\beta = 1$: 22.4. Hessian at $\beta = 20$: 1,787 ($80\times$). Test accuracy: 0.8619 at all β .

2,000 training points. Net 0 (small-batch, test=0.9050). Hessian at $\beta = 1$: 61.9. Hessian at $\beta = 20$: 5,789 ($93\times$). Test accuracy: 0.9050 at all β . Net 25 (large-batch, test=0.9160). Hessian at $\beta = 1$: 24.5. Hessian at $\beta = 20$: 1,671 ($68\times$). Test accuracy: 0.9160 at all β .

6,000 training points. See Table A1 for Net 0. Net 25 (large-batch, test=0.9476). Hessian at $\beta = 1$: 23.5. Hessian at $\beta = 20$: 1,328 ($56\times$). Test accuracy: 0.9476 at all β . Net 30 (large-batch, test=0.9498). Hessian at $\beta = 1$: 22.6. Hessian at $\beta = 20$: 1,318 ($58\times$). Test accuracy: 0.9498 at all β .

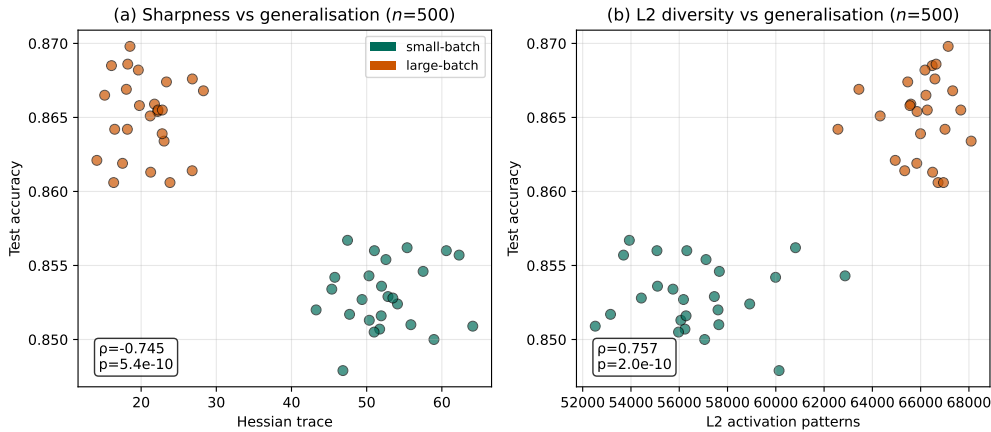


Figure A2. Scatter plots of test accuracy against (a) Hessian trace and (b) L2 activation pattern count at $n = 500$. Both correlate with generalisation at similar strength. Only L2 is reparameterisation-invariant.

Appendix H. Weakness for Frozen-Partition ReLU Networks

This appendix constructs an approximate vocabulary for a ReLU network with frozen early layers and derives the weakness of the training-label policy under that vocabulary. The approximation is exact when each L2 region contains at most one data point and inexact otherwise (see Section H.5 for details).

Appendix H.1. Setup

Fix a 3-layer ReLU MLP $f_{\theta}(x) = W_3\sigma(W_2\sigma(W_1x + b_1) + b_2) + b_3$ where σ is the ReLU activation. Partition the parameters as $\theta = (\theta_{1:2}, \theta_3)$ where $\theta_{1:2} = (W_1, b_1, W_2, b_2)$ are the first two layers and $\theta_3 = (W_3, b_3)$ is the third.

Freeze $\theta_{1:2}$. This fixes the L2 partition. The function $h(x) = \sigma(W_2\sigma(W_1x + b_1) + b_2)$ maps each input x to a D_2 -dimensional hidden representation. The binary activation pattern $\mathbf{1}[W_2\sigma(W_1x + b_1) + b_2 > 0]$ determines a *region* of input space within which h is affine in x . Within a single region, the output $W_3h(x) + b_3$ is also affine in x , so different inputs in the same region can in general receive different classifications. The region-class vocabulary defined below is therefore an *approximation* that treats each region as assigning a single class. This is exact when there is at most one data point per region and approximate otherwise. In all experiments reported here, multiple data points share L2 regions, so the approximation is inexact.

Let $\mathcal{R} = \{r_1, \dots, r_R\}$ be the set of L2 regions that contain at least one data point (training or unseen).

Appendix H.2. The Vocabulary

Definition A1 (Region-class vocabulary). Define the environment $\Phi = \{0, \dots, 9\}^{\mathcal{R}}$, the set of all total classification functions from regions to classes. For each region $r \in \mathcal{R}$ and class $c \in \{0, \dots, 9\}$, define the program

$$p_{r,c} = \{g \in \Phi : g(r) = c\}.$$

The vocabulary is $\mathbf{v}_{\text{net}} = \{p_{r,c} : r \in \mathcal{R}, c \in \{0, \dots, 9\}\}$.

Each program says “region r is classified as c .” For a given region, the 10 programs $p_{r,0}, \dots, p_{r,9}$ are mutually exclusive. No region can be assigned two classes. This is the satisfiability constraint that makes $L_{\mathbf{v}_{\text{net}}} \neq 2^{\mathbf{v}_{\text{net}}}$.

Proposition A3 (Structure of the embodied language). *The embodied language $L_{\mathbf{v}_{\text{net}}}$ is the set of all partial functions from $\mathcal{R}$ to $\{0, \dots, 9\}$, encoded as consistent subsets of $\mathbf{v}_{\text{net}}$. Its cardinality is $|L_{\mathbf{v}_{\text{net}}}| = 11^R$ (10 classes plus “unspecified” for each of R regions).*

Proof. A subset $l \subseteq \mathbf{v}_{\text{net}}$ is consistent (i.e., $\bigcap_{p \in l} p \neq \emptyset$) if and only if no region is assigned two distinct classes. This is exactly the condition for l to encode a partial function from $\mathcal{R}$ to $\{0, \dots, 9\}$. Each of the R regions is either assigned to one of 10 classes or left unspecified, giving 11^R elements. This count includes the empty set $\emptyset \in L_{\mathbf{v}_{\text{net}}}$ (no region classified). $\square$

Appendix H.3. Policy and Weakness

Let $X_{\text{train}} = \{x_1, \dots, x_n\}$ be the training inputs with labels $y_1, \dots, y_n$. Let $\mathcal{R}_{\text{train}} \subseteq \mathcal{R}$ be the set of regions containing at least one training point. For each $r \in \mathcal{R}_{\text{train}}$, let y_r be the majority training label among points in r . When the region contains a single training point, y_r is that point’s label. When it contains multiple training points with possibly different labels, the region-class vocabulary is an approximation. In the model-selection experiments ($784 \rightarrow 64 \rightarrow 16 \rightarrow 10$, 250 training points), many regions contain multiple points, so the approximation is inexact. The pair proxy experiment (Section 5.7) avoids this issue by using per-input linear feasibility rather than the region-class vocabulary.

Definition A2 (Training-label policy). *The training-label policy is $\pi = \{p_{r,y_r} : r \in \mathcal{R}_{\text{train}}\}$. It assigns each training region to the label y_r and is silent on all other regions. When a region contains training points with conflicting labels, y_r is the majority label and the policy is an approximation of the network’s actual behaviour in that region.*

Definition A3 (Free regions). *Let $\mathcal{R}_{\text{free}} = \mathcal{R} \setminus \mathcal{R}_{\text{train}}$ be the set of L2 regions containing unseen data but no training data. Write $k = |\mathcal{R}_{\text{free}}|$.*

These are the regions where the network has committed (it classifies inputs landing there) but the training data provides no constraint on what the classification should be. The network’s behaviour in these regions is determined by the specific setting of θ_3 , not by the child task.

Theorem A2 (Weakness of the training-label policy). *Under the vocabulary $\mathbf{v}_{\text{net}}$, the weakness of the training-label policy π is*

$$w(\pi) = |\text{Ext}(\pi)| = 11^k,$$

where $k = |\mathcal{R}_{\text{free}}|$ is the number of free regions.

Proof. The extension $\text{Ext}(\pi) = \{l \in L_{\mathbf{v}_{\text{net}}} : \pi \subseteq l\}$ consists of all partial classifications that agree with π on training regions. For each training region $r \in \mathcal{R}_{\text{train}}$, the class is fixed to y_r . For each free region $r \in \mathcal{R}_{\text{free}}$, there are 11 independent choices (assign to one of 10 classes, or leave unspecified). So $|\text{Ext}(\pi)| = 11^k$. $\square$

Corollary A1 (Weakness ranking by free regions). *Consider two frozen-partition networks that both memorise the same training data. Let k_1 and k_2 be their respective numbers of free L2 regions. If $k_1 > k_2$, then $w(\pi_1) > w(\pi_2)$.*

Proof. $11^{k_1} > 11^{k_2}$ if and only if $k_1 > k_2$. $\square$

The quantity $k = |\mathcal{R}_{\text{free}}|$ is the “unseen-only regions” measure from the experiments in Section 5. It is reparameterisation-invariant by Theorem 5, because the L2 activation patterns do not change under function-preserving reparameterisation T_β or T_γ .

Appendix H.4. Connection to Generalisation

The sufficiency theorem (Theorem 1) and the finite sufficiency proposition (Proposition A1) predict that policies with higher weakness generalise better. Corollary A1 says that networks with more free regions have higher weakness. Combining these yields the prediction that networks with more free regions generalise better.

There is a subtlety. The sufficiency theorem compares policies *within the same vocabulary*. Two networks with different L2 partitions define different region-class vocabularies. In the strictest reading, the theorem does not directly compare policies across different vocabularies.

The experimental test in Section 5 bypasses this subtlety. It measures k_{free} for each network and correlates it with test accuracy across 300 networks. The result ($\rho = +0.117$, $p = 0.043$) confirms that the prediction holds empirically even across vocabularies. The pair proxy experiment in Section 5.7 provides a stronger test ($\rho = +0.374$, $p = 0.00012$) using linear feasibility rather than region counting.

Appendix H.5. Approximation Caveats

The single-class-per-region approximation. The region-class vocabulary assigns one class per region. In reality, $W_3h(x) + b_3$ is affine in x within a region, so different inputs in the same region can be classified differently. The vocabulary would be exact if each region contained at most one data point. In all experiments reported here, multiple data points share regions, so the approximation is inexact. The pair proxy experiment (Section 5.7) avoids this issue entirely by checking per-input linear feasibility rather than assigning one class per region.

Appendix H.6. The Shared-Weight Caveat

Theorem A2 counts the extension in the region-class vocabulary $\mathbf{v}_{\text{net}}$, where each free region’s classification is an independent choice. But the actual network shares a single weight matrix W_3 across all regions. A setting of W_3 that correctly classifies training regions also determines the classification of free regions. The achievable total classifications of free regions are the cells of a hyperplane arrangement in the polytope of valid W_3 values. These cells are not independent across regions.

This means the region-class vocabulary overstates the freedom of the network. The feature-classifier vocabulary (Appendix I) solves this problem. It defines programs over the shared parameter space rather than per-region classifications, so every element of its embodied language is realisable by a single (W, b) . The pair proxy experiment (Section 5.7) uses this vocabulary. The pair proxy computes single-point extensions exactly (Theorem A3). It is a sum of marginals, not the full extension count $|\text{Ext}(\pi)|$. Computing the full extension exactly remains intractable.

Appendix I. The Feature-Classifier Vocabulary

The region-class vocabulary (Definition A1) approximates the network by assigning one class per region and treating regions as independent. Both are false in general. The shared weight matrix W_3 couples the regions, and multiple inputs per region can receive different classes. The pair proxy experiment avoids both approximations by checking per-input linear feasibility. This appendix proves that the pair proxy computes single-point extensions of the training policy exactly in a well-defined vocabulary that respects the shared-weight constraint.

Fix a feature map $\varphi: X \rightarrow \mathbb{R}^d$ obtained by freezing all layers up to and including a ReLU activation. For an MLP with frozen layers 1 and 2, $\varphi(x) = \sigma(W_2\sigma(W_1x + b_1) + b_2)$. For a CNN with a frozen bottleneck, $\varphi(x) = \sigma(W_{\text{neck}} \text{conv}(x) + b_{\text{neck}})$. In both cases, φ is fixed and the final classifier is $\arg \max_c (W_c \cdot \varphi(x) + b_c)$ with free parameters $(W, b) \in \mathbb{R}^{Kd+K}$.

Let $S \subseteq X$ be a finite set of inputs (training plus unseen probe points) and K the number of classes.

Definition A4 (Feature-classifier vocabulary). Define the environment $\Phi = \mathbb{R}^{Kd+K}$, the set of all parameter settings (W, b) for the final linear layer. For each input $x \in S$ and class $c \in \{0, \dots, K-1\}$, define the program

$$p_{x,c} = \{(W, b) \in \Phi : W_c \cdot \varphi(x) + b_c > W_{c'} \cdot \varphi(x) + b_{c'} \text{ for all } c' \neq c\}.$$

The feature-classifier vocabulary is $\mathbf{v}_\varphi = \{p_{x,c} : x \in S, c \in \{0, \dots, K-1\}\}$.

Each program says “input x is classified as c by the final linear layer, given the frozen features $\varphi(x)$.” Each $p_{x,c}$ is an open subset of $\mathbb{R}^{Kd+K}$ defined by finitely many strict linear inequalities. For fixed x , the K programs $p_{x,0}, \dots, p_{x,K-1}$ are pairwise disjoint. No parameter vector can assign x to two classes simultaneously.

The embodied language $L_{\mathbf{v}_\varphi}$ is the set of all consistent subsets $l \subseteq \mathbf{v}_\varphi$, where consistency means $\bigcap_{p \in l} p \neq \emptyset$. Some parameter setting satisfies all programs in l simultaneously. This follows the standard Stack Theory definition of $L_{\mathbf{v}}$. The pairwise disjointness of $p_{x,0}, \dots, p_{x,K-1}$ ensures that every $l \in L_{\mathbf{v}_\varphi}$ assigns at most one class per input.

Proposition A4 (Consistency is LP feasibility). A subset $l \subseteq \mathbf{v}_\varphi$ is in $L_{\mathbf{v}_\varphi}$ if and only if the following system of strict linear inequalities is feasible. For each $p_{x_i,c_i} \in l$ and each $c \neq c_i$,

$$W_{c_i} \cdot \varphi(x_i) + b_{c_i} > W_c \cdot \varphi(x_i) + b_c.$$

Equivalently, $l \in L_{\mathbf{v}_\varphi}$ if and only if for some $\varepsilon > 0$, the LP with constraints $(W_{c_i} - W_c) \cdot \varphi(x_i) + (b_{c_i} - b_c) \geq \varepsilon$ for all $p_{x_i,c_i} \in l$ and $c \neq c_i$ is feasible.

Proof. By Definition A4, $\bigcap_{p \in l} p$ is the set of all (W, b) satisfying the strict inequalities for every program in l . If l assigns two classes to the same input (say p_{x,c_1} and p_{x,c_2} with $c_1 \neq c_2$), the system contains $W_{c_1} \cdot \varphi(x) + b_{c_1} > W_{c_2} \cdot \varphi(x) + b_{c_2}$ and its reverse, which is infeasible. Otherwise, $l \in L_{\mathbf{v}_\varphi}$ if and only if the conjunction of strict inequalities has a solution.

For the LP equivalence, suppose (W^*, b^*) satisfies all strict inequalities. Define $\varepsilon^* = \min_{i,c \neq c_i} [(W_{c_i}^* - W_c^*) \cdot \varphi(x_i) + (b_{c_i}^* - b_c^*)] > 0$. The minimum is over finitely many strictly positive terms, so $\varepsilon^* > 0$. Then (W^*, b^*) is LP-feasible for any $\varepsilon \leq \varepsilon^*$. Conversely, any LP solution with $\varepsilon > 0$ satisfies all strict inequalities. $\square$

Checking whether a set of input-class assignments is simultaneously achievable by some linear classifier is an LP feasibility problem. The LP does not approximate the vocabulary. It checks consistency in $\mathbf{v}_\varphi$ exactly, provided the margin ε is small enough to not exclude feasible solutions. In the Fashion-MNIST replication, the adaptive margin procedure finds the maximum feasible ε and uses half of it, guaranteeing this condition. In the original MNIST pair-proxy experiment, the LP instead uses a fixed small margin, which serves the same role provided it is chosen below the feasible maximum margin.

Definition A5 (Training policy in $\mathbf{v}_\varphi$). Given training data $(x_1, y_1), \dots, (x_n, y_n)$ with $x_i \in S$, the training policy is

$$\pi = \{p_{x_i,y_i} : i = 1, \dots, n\}.$$

Assume $\pi \in L_{\mathbf{v}_\varphi}$. This holds whenever the trained parameters (W^*, b^*) classify every training input x_i as y_i with a strict margin, because then $(W^*, b^*) \in \bigcap_{p \in \pi} p$.

Definition A6 (Pair proxy as extension count). For an unseen input $x_j \in S \setminus \{x_1, \dots, x_n\}$, the pointwise extension count is

$$\text{ext}(x_j) = |\{c \in \{0, \dots, K-1\} : \pi \cup \{p_{x_j,c}\} \in L_{\mathbf{v}_\varphi}\}|.$$

The pair proxy is

$$\text{PP}(\varphi) = \sum_j \text{ext}(x_j).$$

Each term $\text{ext}(x_j)$ counts the number of classes to which x_j can be assigned while keeping the training constraints satisfiable by some shared (W, b) . By Proposition A4, each such check is an LP feasibility problem.

The pair proxy is not the full weakness $|\text{Ext}(\pi)|$. The full weakness counts all consistent partial classifications extending π , including multi-point extensions and partial assignments. The pair proxy counts single-point extensions, summed across unseen inputs. Computing $|\text{Ext}(\pi)|$ exactly would require enumerating an exponential number of joint assignments. The pair proxy is a tractable quantity derived from the single-point marginals of the extension set.

Theorem A3 (Feature-classifier reduction). *Fix a feature map φ , training data (x_i, y_i) with $\pi \in L_{\mathbf{v}_\varphi}$, and the feature-classifier vocabulary $\mathbf{v}_\varphi$.*

1. For each unseen x_j , $\text{ext}(x_j) = |\{c : p_{x_j,c} \text{ appears in some } l \in \text{Ext}(\pi)\}|$. The pointwise extension count equals the number of classes that appear in at least one completion of the training policy.
2. $\text{PP}(\varphi)$ is computable by solving $K \cdot |S_{\text{unseen}}|$ LP feasibility problems.
3. PP is invariant under invertible affine reparameterisation of the feature space. Let $A \in \mathbb{R}^{d \times d}$ be invertible and define $\varphi'(x) = A \varphi(x)$. Then $\text{PP}(\varphi') = \text{PP}(\varphi)$. In particular, PP is invariant under the scaling reparameterisation T_β (where $A = \beta I$).
4. $\mathbf{v}_\varphi$ respects the shared-weight constraint by construction. Every $l \in L_{\mathbf{v}_\varphi}$ is realisable by some single (W, b) . The region-class vocabulary does not have this property.

Proof. Claim 1. I prove both directions.

($\Leftarrow$) Suppose $\pi \cup \{p_{x_j,c}\}$ is consistent. Then $\pi \cup \{p_{x_j,c}\} \in L_{\mathbf{v}_\varphi}$. Since $\pi \subseteq \pi \cup \{p_{x_j,c}\}$, this set is in $\text{Ext}(\pi)$ and contains $p_{x_j,c}$.

($\Rightarrow$) Suppose $p_{x_j,c} \in l$ for some $l \in \text{Ext}(\pi)$. Then $\pi \cup \{p_{x_j,c}\} \subseteq l$ (because $\pi \subseteq l$ and $p_{x_j,c} \in l$). Since l is consistent and consistency is inherited by subsets ($l' \subseteq l$ implies $\bigcap_{p \in l'} p \supseteq \bigcap_{p \in l} p \neq \emptyset$), $\pi \cup \{p_{x_j,c}\}$ is consistent.

Claim 2. By Proposition A4, each consistency check $\pi \cup \{p_{x_j,c}\} \in L_{\mathbf{v}_\varphi}$ reduces to LP feasibility. There are K classes and $|S_{\text{unseen}}|$ unseen inputs.

Claim 3. Let $\varphi'(x) = A \varphi(x)$ for invertible A , and let $\mathbf{v}_{\varphi'}$ be the corresponding vocabulary. Define the bijection $\Psi: \mathbb{R}^{Kd+K} \rightarrow \mathbb{R}^{Kd+K}$ by $\Psi(W, b) = (WA^{-1}, b)$. For any $x \in S$ and class c ,

$$\begin{aligned} (W, b) \in p_{x,c} &\iff W_c \cdot \varphi(x) + b_c > W_{c'} \cdot \varphi(x) + b_{c'} \text{ for all } c' \neq c \\ &\iff (WA^{-1})_c \cdot A \varphi(x) + b_c > (WA^{-1})_{c'} \cdot A \varphi(x) + b_{c'} \text{ for all } c' \neq c \\ &\iff \Psi(W, b) \in p'_{x,c}, \end{aligned}$$

where $p'_{x,c}$ is the program for x, c in $\mathbf{v}_{\varphi'}$. The second line uses $WA^{-1} \cdot A \varphi(x) = W \cdot \varphi(x)$.

So Ψ maps $p_{x,c}$ bijectively to $p'_{x,c}$ for every $x \in S$ and c . It follows that $l \in L_{\mathbf{v}_\varphi}$ if and only if $\Psi(l) \in L_{\mathbf{v}_{\varphi'}}$ (where $\Psi(l)$ replaces each program by its image). Since Ψ maps π to π' and preserves consistency, $\text{ext}(x_j)$ is the same under φ and φ' , and $\text{PP}(\varphi) = \text{PP}(\varphi')$.

Claim 4. By definition, $l \in L_{\mathbf{v}_\varphi}$ means $\bigcap_{p \in l} p \neq \emptyset$. Each $p \in l$ is a subset of $\Phi = \mathbb{R}^{Kd+K}$, so a nonempty intersection is a parameter setting (W, b) satisfying all constraints in l simultaneously. In contrast, the region-class vocabulary defines $\Phi = \{0, \dots, 9\}^R$ and consistency means a compatible partial function exists, regardless of whether any single (W, b) realises it. $\square$

The key distinction from the region-class vocabulary is Claim 4. The region-class vocabulary has $|L_{\mathbf{v}_{\text{net}}}| = 11^R$ and overstates the extension because it treats each region's classification as an independent degree of freedom. In the feature-classifier vocabulary, every element of $L_{\mathbf{v}_\varphi}$ must be

realisable by a single shared parameter vector. This is exactly the shared-weight vocabulary identified in Section H.6.

Corollary A2 (Architecture-independence). *Let φ_1 and φ_2 be two frozen feature maps (from potentially different architectures) that produce the same feature vectors on all inputs in S . That is, $\varphi_1(x) = \varphi_2(x)$ for all $x \in S$. Then $\mathbf{v}_{\varphi_1} = \mathbf{v}_{\varphi_2}$ and $\text{PP}(\varphi_1) = \text{PP}(\varphi_2)$.*

Proof. The programs $p_{x,c}$ depend on φ only through the values $\varphi(x)$ for $x \in S$. Equal feature vectors produce identical programs, hence identical vocabularies, languages, extensions, and pair proxies. $\square$

The pair proxy depends only on the feature representation, not on the architecture that produced it. An MLP with hidden representation $\varphi(x) \in \mathbb{R}^8$ and a CNN with an 8-dimensional bottleneck define the same vocabulary if their features agree on S . For a bottleneck width of $d = 8$ and $K = 10$ classes, the LP has $Kd + K = 90$ variables. This matches the scale of the MNIST and Fashion-MNIST pair proxy experiments ($784 \rightarrow 64 \rightarrow 8 \rightarrow 10$ gives $d = 8$).

References

1. Hochreiter, S.; Schmidhuber, J. Flat Minima. *Neural Computation* **1997**, *9*, 1–42. <https://doi.org/10.1162/neco.1997.9.1.1>.
2. Keskar, N.S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; Tang, P.T.P. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In Proceedings of the International Conference on Learning Representations, 2017.
3. Foret, P.; Kleiner, A.; Mobahi, H.; Neyshabur, B. Sharpness-aware Minimization for Efficiently Improving Generalization. In Proceedings of the International Conference on Learning Representations, 2021.
4. Dinh, L.; Pascanu, R.; Bengio, S.; Bengio, Y. Sharp Minima Can Generalize For Deep Nets. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning. PMLR, 2017, Vol. 70, *Proceedings of Machine Learning Research*, pp. 1019–1028.
5. Tsuzuku, Y.; Sato, I.; Sugiyama, M. Normalized Flat Minima: Exploring Scale Invariant Definition of Flat Minima for Neural Networks Using PAC-Bayesian Analysis. In Proceedings of the Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020, Vol. 119, *Proceedings of Machine Learning Research*, pp. 9636–9647.
6. Petzka, H.; Kamp, M.; Adilova, L.; Sminchisescu, C.; Boley, M. Relative Flatness and Generalization. In Proceedings of the Advances in Neural Information Processing Systems, 2021, Vol. 34, pp. 18420–18432.
7. Kwon, J.; Kim, J.; Park, H.; Choi, I.K. ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning. PMLR, 2021, Vol. 139, *Proceedings of Machine Learning Research*, pp. 5905–5914.
8. Bennett, M.T. The Optimal Choice of Hypothesis Is the Weakest, Not the Shortest. In Proceedings of the 16th International Conference on Artificial General Intelligence. Springer, 2023, Lecture Notes in Computer Science, pp. 42–51. https://doi.org/10.1007/978-3-031-33469-6_5.
9. Bennett, M.T. How To Build Conscious Machines. PhD thesis, The Australian National University, 2025. <https://doi.org/10.25911/TA58-P428>.
10. Bennett, M.T. Regret Is Weighted Forgetting, 2026.
11. Bennett, M.T. Technical Appendices, 2025. Archived release on Zenodo. Source repository: <https://github.com/ViscousLemming/Technical-Appendices>, <https://doi.org/10.5281/zenodo.7641741>.
12. Bennett, M.T. Is Complexity an Illusion? In Proceedings of the 17th International Conference on Artificial General Intelligence. Springer, 2024, Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-031-65572-2_2.
13. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017, [arXiv:cs.LG/1708.07747].
14. Vapnik, V.N.; Chervonenkis, A.Y. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and Its Applications* **1971**, *16*, 264–280. <https://doi.org/10.1137/1116025>.
15. Bousquet, O.; Elisseeff, A. Stability and Generalization. *Journal of Machine Learning Research* **2002**, *2*, 499–526. <https://doi.org/10.1162/153244302760200704>.

16. McAllester, D.A. PAC-Bayesian Model Averaging. In Proceedings of the Proceedings of the Twelfth Annual Conference on Computational Learning Theory. ACM, 1999, pp. 164–170. <https://doi.org/10.1145/307400.307435>.
17. Rissanen, J. Modeling by Shortest Data Description. *Automatica* **1978**, *14*, 465–471. [https://doi.org/10.1016/0005-1098\(78\)90005-5](https://doi.org/10.1016/0005-1098(78)90005-5).
18. Hutter, M. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*; Springer: Berlin, 2005. <https://doi.org/10.1007/b138233>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.