Understanding Deep Learning Requires Rethinking Generalization

Zhang, Bengio, Hardt, Recht & Vinyals (ICLR 2017)

Theory of Generalization Reading Group

Where we are

Last time: the **classical** picture (Shalev-Shwartz & Ben-David).

The chain we built

$$\text{VCdim} < \infty \iff \text{uniform convergence} \iff \text{PAC learnable} \iff \text{ERM succeeds}$$

Moral of that deck: **generalization comes from controlling the complexity of the hypothesis class**.

Today: a paper whose entire purpose is to show that this moral, taken literally, *cannot explain* why deep networks generalize — and to make the failure precise with experiments and one clean theorem.

Outline

Recap: the classical definitions

The puzzle and the paper's claims

Randomization tests: results

Why the classical measures fail

The role of regularization (results)

Finite-sample expressivity (the theorem)

Using linear models as a toy setting to study implicit bias

Synthesis

Recap: the classical definitions

The two risks and ERM (reminder)

Unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$; sample $S = \{(x_i, y_i)\}_{i=1}^m \sim \mathcal{D}^m$.

True risk (what we want)

$$L_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}\big[h(x) \neq y\big]$$

Defined on $\mathcal{D}$. *Cannot* be computed.

Empirical risk (what we have)

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} 1[h(x_i) \neq y_i]$$

Defined on the sample S only.

ERM: $h_{\mathrm{ERM}} = \arg\min_{h\in\mathcal{H}} L_S(h)$ — fit the training data as well as possible.

Key fact: for a *fixed* h, $\mathbb{E}[L_S(h)] = L_{\mathcal{D}}(h)$ and it concentrates (Hoeffding). The whole theory is about surviving the fact that ERM *chooses* h using S.

The complexity measures we will put on trial

The paper's targets are exactly the classical capacity notions:

- **VC dimension**: largest set $\mathcal{H}$ can *shatter* (realise all $2^{|C|}$ labelings). A property of the *class* alone.
- **(Empirical) Rademacher complexity** on $\{x_1, \ldots, x_n\}$:

$$\widehat{\mathfrak{R}}_n(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \, h(x_i) \right], \qquad \sigma_i \overset{\text{iid}}{\sim} \mathrm{Unif}\{\pm 1\}.$$

  Literally: how well $\mathcal{H}$ can fit *random* ± 1 *labels*.
- **Uniform stability**: sensitivity of the *algorithm* to swapping one training example.

Hold onto the Rademacher definition: the paper's main experiment is a Rademacher complexity measurement in disguise.

The classical promise — and its hidden quantifier

Fundamental Theorem of Statistical Learning

For binary classification with 0–1 loss, the following are equivalent: finite VC dimension, uniform convergence, agnostic PAC learnability, and the success of *any* ERM rule.

Two things to underline before we proceed:

- The guarantee is **distribution-free**: it must hold for *every* $\mathcal{D}$.
- Complexity is attributed to $\mathcal{H}$; the **algorithm is interchangeable** ("any ERM rule").

Both of these are precisely what deep networks will quietly escape.

The puzzle and the paper's claims

The puzzle

Modern networks have:

- millions–billions of parameters; effectively **enormous VC dimension**;
- the ability to **fit random labels** (i.e. shatter the training set);
- near-perfect interpolation of the training data.

Yet test error is low, and scaling often *improves* generalization.

Central question of the paper

What distinguishes networks that generalize from those that don't, if the classical complexity measures are blind to the difference?

Contributions of the paper

1. **Randomization tests.** Standard nets reach 0 training error on *random labels* — and even on *random pixels*.
2. **Regularization.** Explicit regularization is *neither necessary (removing regularizers doesn't remove generalization) nor sufficient (training on random labels + regularization doesn't imply generalization* for generalization.
3. **Finite-sample expressivity (Theorem 1).** A depth-2 ReLU net with $2n + d$ weights can represent *any* labeling of n points in $\mathbb{R}^d$.
4. **Implicit regularization.** Via linear models: SGD converges to *minimum-norm* solutions (but even that is not predictive of generalization).

Randomization tests: results

Methodology: the randomization test

Take a fixed architecture, fixed hyperparameters, fixed optimizer. Train twice:

True labels

ordinary learning problem

Random labels

labels reassigned uniformly at random; *no* relationship to inputs $\Rightarrow$ learning is impossible

Variants on the *inputs* too: shuffled pixels, per-image random pixels, Gaussian noise.

Intuition says random labels should break training (no convergence, huge slowdown). The finding is that it doesn't.

Main finding

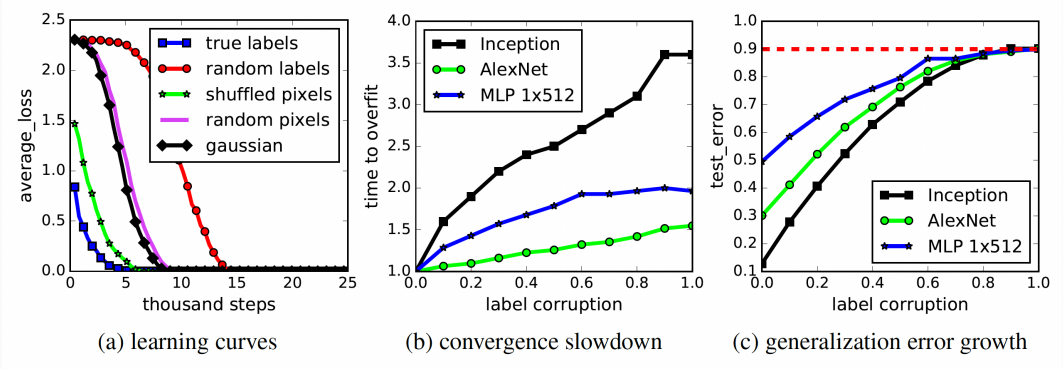
Deep neural networks easily fit random labels.

Three implications the authors draw:

1. The **effective capacity** of these nets suffices to *memorize the entire dataset*.
2. **Optimization stays easy:** training time grows only by a small constant factor.
3. Randomizing labels is *only a data transformation* — model, size, hyperparameters, optimizer all unchanged.

So: by changing nothing but the labels, generalization error jumps from small to chance. Whatever explained the small error before must not have been a property of the model class.

Experimental evidence



(a) learning curves (b) convergence slowdown (c) generalization error growth

Changing only the labels transforms good generalization into pure memorization.

Breaking the inputs, not just the labels

Convolutional nets still reach zero training error when the *images* are destroyed:

- **shuffled pixels** (one fixed permutation), **random pixels** (per image), **Gaussian noise**.

Sweeping the corruption level p from 0 to 1 interpolates between the real problem and pure noise:

- the net fits the corrupted set *perfectly* at every p;
- test (= generalization) error rises *smoothly* with p, reaching $\approx 90\%$ (chance on CIFAR-10) at $p = 1$.

Reading: the network simultaneously *captures* whatever signal remains and *brute-force memorizes* the noise.

The numbers (CIFAR-10, Table 1)

Model (no regularization)	#params	train acc.	test acc.
Inception (true labels)	1.65M	100.0	85.75
Inception (**random labels**)	1.65M	100.0	9.78
Alexnet (true labels)	1.39M	100.0	76.07
Alexnet (**random labels**)	1.39M	99.82	9.86
MLP 1×512 (true labels)	1.21M	100.0	50.51
MLP 1×512 (**random labels**)	1.21M	99.34	10.61

ImageNet (Inception v3): 95.2% *train* accuracy on 1.28M **random** labels across 1000 classes (test accuracy $\approx 0.1\%$, i.e. chance).

Same architecture, same optimizer — only the labels differ. Train accuracy $\approx 100\%$ either way; test accuracy collapses to chance.

Why the classical measures fail

What this does to the classical picture

$$\underbrace{\text{Hypothesis class}}_{\text{same in both experiments}} \overset{?}{\Longrightarrow} \text{generalization}$$

The randomization test keeps the model class fixed, yet changing only the labels transforms

$$\text{generalization} \quad \longrightarrow \quad \text{memorization.}$$

Conclusion

The explanation cannot depend only on the capacity of $\mathcal{H}$.

The missing ingredient must come from the data and/or the training process.

The role of regularization (results)

Explicit regularization: neither necessary nor sufficient

Explicit regularization may improve generalization, but is *neither necessary nor by itself sufficient* for controlling generalization error.

Evidence (CIFAR-10 / ImageNet):

- Turn *all* of weight decay / dropout / augmentation **off** $\Rightarrow$ models *still generalize well* (e.g. Inception 85.75% test, vs 89% with everything on).
- Most architectures *still fit random labels* even with weight decay *on*.
- On ImageNet, data augmentation helps more than weight decay/dropout.

Unlike classical convex ERM, where regularization controls the effective hypothesis class, deep networks often generalize even in its absence.

Finite-sample expressivity (the theorem)

From function approximation to interpolation

Classical theory asks which functions can be represented on the whole input space. Zhang et al. ask a weaker but crucial question: can a network interpolate an arbitrary finite dataset, even with random labels?

Definition (Representing a sample)

A network C represents *any* function of a sample of size n in d dimensions if for every $S \subseteq \mathbb{R}^d$ with $|S| = n$ and every $f : S \to \mathbb{R}$, there exist weights with $C(x) = f(x)$ for all $x \in S$.

This is exactly "can shatter the sample": the formal counterpart of the random-label experiments.

Theorem 1: Finite-sample expressivity

Theorem (Finite-sample expressivity)

There exists a two-layer ReLU network with $2n + d$ weights that can represent any function on a sample of size n in d dimensions.

Interpretation

The randomization experiment is not an optimization accident: generic overparameterized ReLU networks can interpolate *arbitrary* labelings of a finite dataset.

Lemma

For interleaving reals $b_1 < x_1 < b_2 < x_2 < \cdots < b_n < x_n$, the matrix

$$A = [\max\{x_i - b_j, 0\}]_{ij}$$

has full rank, with smallest eigenvalue $\min_i(x_i - b_i)$.

Proof of the lemma

Claim: $A = [\max\{x_i - b_j, 0\}]_{ij}$ is lower-triangular and invertible.

- For $i < j$: the interleaving order gives $x_i < b_j$, so $x_i - b_j < 0$ and $\max\{x_i - b_j, 0\} = 0$. Hence all entries *above* the diagonal vanish $\Rightarrow$ A is lower-triangular.
- On the diagonal: $x_i > b_i$, so $\max\{x_i - b_i, 0\} = x_i - b_i > 0$.
- A triangular matrix has its eigenvalues on the diagonal and is invertible iff all diagonal entries are nonzero. All are positive $\Rightarrow$ full rank, smallest eigenvalue $\min_i(x_i - b_i)$.

□

Proof of Theorem 1

Use the depth-2 ReLU function, with $a \in \mathbb{R}^d$, $b, w \in \mathbb{R}^n$:

$$c(x) = \sum_{j=1}^{n} w_j \max\{\langle a, x\rangle - b_j,\ 0\}.$$

Given sample $S = \{z_1, \ldots, z_n\}$ and target $y \in \mathbb{R}^n$, find a, b, w with $c(z_i) = y_i$.

1. **Pick a direction.** Choose a so the projections $x_i = \langle a, z_i\rangle$ are distinct; choose b so that $b_1 < x_1 < \cdots < b_n < x_n$ *interleave*. (Possible since the z_i are distinct.)
2. **Reduce to a linear system.** Then $c(z_i) = (Aw)_i$ with A the lemma's matrix. The n equations $y = Aw$ have a solution because, by the lemma, A is invertible: $w = A^{-1}y$. □

Parameter count: a contributes d, and b, w contribute n each $\Rightarrow 2n + d$. A depth-k variant (Corollary 1) trades width for depth: width $O(n/k)$, $O(n + d)$ weights.

Using linear models as a toy setting to study implicit bias

Even linear models are subtle

Data $\{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^d$; minimize empirical risk $\min_w \frac{1}{n} \sum_i \mathrm{loss}(w^\top x_i, y_i)$.

If $d \geq n$ and X (the $n \times d$ data matrix) has rank n, then $Xw = y$ has *infinitely many* solutions. You can fit any labeling.

- **Curvature doesn't help:** the Hessian $\frac{1}{n} X^\top \mathrm{diag}(\beta) X$ is the *same at every global minimum* (and degenerate). So sharpness/flatness cannot distinguish the good solutions from the bad here.

So which of the infinitely many interpolating solutions do we get? SGD is the answer.

SGD selects the minimum-norm interpolant

SGD update: $w_{t+1} = w_t - \eta_t e_t x_{i_t}$ (with e_t the prediction error).

- Starting from $w_0 = 0$, every update adds a multiple of some x_{i_t}, so the iterate stays in the **span of the data**: $w = X^\top \alpha = \sum_i \alpha_i x_i$.
- Interpolation requires $Xw = y$. Substituting:

$$\boxed{XX^\top \alpha = y} \qquad \text{(a unique solution; depends only on inner products } x_i^\top x_j\text{)}.$$

This is the **kernel / Gram-matrix** solution $K\alpha = y$, $K = XX^\top$ — and it is exactly the *minimum ℓ_2-norm* solution of $Xw = y$. SGD is, in this sense, *implicitly regularizing*.

Interpolation $\neq$ overfitting

Fitting labels *exactly* with the min-norm kernel solution generalizes surprisingly well (no explicit regularization):

data	pre-processing	test error
MNIST	none	1.2%
MNIST	Gabor wavelets	0.6%
CIFAR-10	none	46%
CIFAR-10	random conv-net	17%

But minimum norm is *not* the whole story

On MNIST the min-norm solution has ℓ_2-norm ≈ 220 (no preprocessing) vs ≈ 390 (with wavelets) — yet the *larger*-norm solution has *half* the test error. So "small norm" does not predict generalization; it is at most a small piece.

Synthesis

Back to the three questions

1. *Why can networks fit random labels?*
   Capacity. Theorem 1: $2n + d$ parameters $\Rightarrow$ can shatter the sample. Fully answered.
2. *Why does interpolation not imply overfitting?*
   The *same* class/optimizer that fits random labels also generalizes on real labels. Capacity is maxed in both cases, so capacity cannot be the explanation — the answer lives in the data + algorithm. (Gestured at via min-norm; not fully resolved.)
3. *Is VC dimension the wrong notion of complexity?*
   It is the wrong *tool* here: VC/Rademacher/stability are distribution-free and (for the first two) class-only. What governs deep-net generalization is *data- and algorithm-dependent*.

What the paper does and doesn't settle

Settles (negatively):

- VC, Rademacher, uniform stability *cannot* explain the observed generalization.
- Explicit regularization is not the mechanism.
- Optimization being easy is *separate* from generalization.

Leaves open (positively):

- *Why* SGD-trained interpolators generalize.
- A formal capacity measure under which these huge models are "simple".
- Min-norm is suggestive but not predictive.

The reframing: the Fundamental Theorem is still *true* — deep nets simply fall outside its (worst-case, distribution-free) hypotheses. Generalization here is a joint property of architecture, data, and optimizer.

Effective capacity is large enough to memorize the data.

The open problem is to find the measure under which these models are nonetheless *simple*.

References i

- C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals. *Understanding Deep Learning Requires Rethinking Generalization.* ICLR 2017. arXiv:1611.03530.
- S. Shalev-Shwartz, S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge Univ. Press, 2014.
- P. Bartlett, S. Mendelson. *Rademacher and Gaussian complexities.* JMLR 2003.
- O. Bousquet, A. Elisseeff. *Stability and generalization.* JMLR 2002.
- B. Neyshabur, R. Tomioka, N. Srebro. *In search of the real inductive bias.* 2014.
- M. Hardt, B. Recht, Y. Singer. *Train faster, generalize better.* ICML 2016.