## A Proof of Theorem 3.1: KR Trichotomy

In this appendix, we provide proofs and further discussion of our theoretical results for KR. To aid readers, we begin by reproducing Equation 3 for the approximate generalization MSE of KR:

$$\mathcal{E} \approx \mathcal{E}_0 \left( \sum_i (1-\mathcal{L}_i)^2 v_i^2 + \varepsilon^2 \right), \quad \text{where} \quad \mathcal{E}_0 \equiv \frac{n}{n - \sum_j \mathcal{L}_j^2}, \\ \mathcal{L}_i \equiv \frac{\lambda_i}{\lambda_i + \kappa}, \quad \text{and} \quad \kappa \geq 0 \text{ satisfies } \sum_i \frac{\lambda_i}{\lambda_i + \kappa} + \frac{\delta}{\kappa} = n. \tag{3}$$

The works obtaining this result [Bordelon et al., 2020, Canatar et al., 2021, Jacot et al., 2020, Simon et al., 2021] make certain approximations common in the statistical physical literature. The main one is the *universality* assumption that the eigenfunctions can be replaced by structureless Gaussian random variables without changing the statistics of test risk (see Jacot et al. [2020] for a formalization of this assumption). Though this step discards most of the information in the problem (leaving only the scalar eigenvalues and eigencoefficients), it is validated by the close match between experiment and the theory derived with this approximation. KR is equivalent to linear regression in the high-dimensional kernel embedding space defined by its eigenfeatures, and we note several recent works which have derived the same equations for high-dimensional linear regression, analogously assuming i.i.d. random covariates [Bartlett et al., 2021, Hastie et al., 2019]. This assumption of i.i.d. random covariates is also made by Bartlett et al. [2020].

In studying these equations, we assume the following:

**Assumption A.1.** *The kernel eigenvalues* $\{\lambda_i\}_{i=1}^{\infty}$ *and target function eigencoefficients* $\{v_i\}_{i=1}^{\infty}$ *satisfy*

*(a)* $\lambda_i > \lambda_j$ *if* $i > j$,

*(b)* $\sum_{i=1}^{\infty} \lambda_i < \infty$,

*(c)* $\{\lambda_i\}_{i=1}^{\infty}$ *contains infinitely many nonzero elements,*

*(d)* $\sum_{i=1}^{\infty} v_i^2 < \infty$, *and*

*(e)* $\lim_{\lambda \to 0^+} \sum_{i | \lambda_i < \lambda} v_i^2 = 0$.

These natural assumptions imply that (a) we have indexed the eigenvalues in descending order, (b) the trace of the kernel is finite, (c) the rank of the kernel is infinite (which is typically the case in practice), (d) the target function has finite $\ell^2$-norm, and (e) the target function does not place non-negligible weight in arbitrarily-low (or zero) eigenmodes[11]. While condition (e) is somewhat nonstandard, we note that it is strictly *weaker* than requiring finite RKHS norm of $f$ w.r.t. $K$:

$$\|f\|_K^2 = \sum_{i=1}^{\infty} \frac{v_i^2}{\lambda_i} < \infty. \tag{4}$$

We note that all eigenvalues are nonnegative because the kernel is positive semidefinite.

The constant $\kappa$ fixed by Equation 3 satisfies

$$\kappa \leq \frac{1}{n\gamma} \left( \delta + \sum_{i \geq n(1-\gamma)} \lambda_i \right), \tag{5}$$

$$\kappa \geq \gamma \lambda_{n(1+\gamma)}, \tag{6}$$

[11] Weight placed in zero eigenmodes (i.e. lying outside the RKHS of the kernel) should be regarded as noise and included in $\varepsilon^2$ instead of $\{v_i\}_i$.

where $\gamma \in (0,1)$ in Equation 5, $\gamma \in (0,\infty)$ in Equation 6, and in both cases $\gamma$ satisfies $\gamma n \in \mathbb{Z}$. These bounds follow from more general bounds in Simon et al. [2021]. We will use them shortly.

**Proof of Theorem 3.1**. We begin by observing from Equation 5 with any $\gamma \in (0,1)$ that $\kappa = \mathcal{O}(1/n)$, and thus $\lim_{n\to\infty} \kappa = 0$ and $\lim_{n\to\infty} \mathcal{L}_i = 1$ if $\lambda_i$ is bounded away from 0. Paired with Assumption A.1e, this implies that $\lim_{n\to\infty} \sum_i (1-\mathcal{L}_i)^2 v_i^2 = 0$, and thus, examining Equation 3, we find that asymptotic MSE is dominated by the noise $\varepsilon^2$ and given by

$$\lim_{n\to\infty} \mathcal{E} = \left(\lim_{n\to\infty} \mathcal{E}_0\right) \varepsilon^2. \tag{7}$$

Asymptotic MSE is thus dominated by the noise, and we can neglect the target coefficients $\{v_i\}_i$.

We now prove each clause of the theorem by finding $\lim_{n\to\infty} \mathcal{E}_0$, beginning with the two parts of clause (a).

***Clause (a):*** *if* $\delta > 0$, *then* $\lim_{n\to\infty} \mathcal{E} = \varepsilon^2$.

We assume that $\delta > 0$. To begin, we note a lower bound for $\kappa$. Since

$$n = \sum_{i=1}^{\infty} \frac{\lambda_i}{\lambda_i + \kappa} + \frac{\delta}{\kappa} \geq \frac{\delta}{\kappa}, \tag{8}$$

it follows that $\kappa \geq \delta/n$. Using this bound, we find that

$$\frac{1}{n}\sum_{i=1}^{\infty} \mathcal{L}_i^2 = \frac{1}{n}\sum_{i=1}^{\infty} \frac{\lambda_i^2}{(\lambda_i+\kappa)^2} \leq \gamma + \frac{1}{n}\sum_{i\geq n\gamma} \frac{\lambda_i^2}{\kappa^2} \leq \gamma + n\sum_{i\geq n\gamma} \frac{\lambda_i^2}{\delta^2}, \tag{9}$$

where $\gamma \in (0,1)$ and we have "given up" on the first $\gamma n$ terms in the sum, replacing them with 1.

Because $\sum_{i=1}^{\infty} \lambda_i < \infty$, it must hold that $\lambda_i = o(i^{-1})$, and thus the sum on the far RHS of Equation 9 approaches zero as $n \to \infty$. This tells us that $\lim_{n\to\infty} \mathcal{E}_0 = \lim_{n\to\infty} n/(n + \sum_i \mathcal{L}_i^2) \leq 1/(1+\gamma)$. By choosing $\gamma$ to be arbitrarily small, we can push this bound to 1, which proves the clause.

***Clause (a'):*** *if* $\lambda_i = i^{-1}\log^{-\alpha} i$ *for some* $\alpha > 1$, *then* $\lim_{n\to\infty} \mathcal{E} = \varepsilon^2$.

We first set $\gamma = 1$ in Equation 6 and find that $\kappa \geq \lambda_{2n} = (2n)^{-1}\log^{-\alpha}(2n)$. Anticipating our next move, we then note that, if $i \geq n$,

$$\kappa i \log^{\alpha} i \geq \frac{\log^{\alpha} n}{2(\log^{\alpha} n + \log^{\alpha} 2)} \geq \frac{1}{5} \quad \text{for sufficiently large } n. \tag{10}$$

We then observe that

$$n = \sum_{i=1}^{\infty} \frac{1}{1+\kappa i \log^{\alpha} i} \geq \sum_{i\geq n} \frac{1}{1+\kappa i \log^{\alpha} i} \geq \frac{1}{6\kappa}\sum_{i\geq n} \frac{1}{i\log^{\alpha} i} \geq \frac{1}{6\kappa}\frac{1}{(\alpha-1)\log^{\alpha-1} n}, \tag{11}$$

where in the third step we have used Equation 10 and in the fourth step we have used the fact that $\int_x^{\infty} z^{-1}\log^{-\alpha} z dz = (\alpha-1)^{-1}\log^{-\alpha+1} x$. This tells us that

$$\kappa \geq \left(6(\alpha-1) n \log^{\alpha-1} n\right)^{-1}. \tag{12}$$

We then observe that

$$\sum_{i=1}^{\infty} \mathcal{L}_i^2 \leq \frac{n}{\log\log n} + \sum_{i\geq \frac{n}{\log\log n}} \frac{\lambda_i^2}{\kappa^2} = \frac{n}{\log\log n} + \kappa^{-2}\Theta\left(\frac{\log\log n}{n\log^{2\alpha} n}\right), \tag{13}$$

where in the first step we have "given up" on the first $\frac{n}{\log\log n}$ terms and in the second step we have used the fact that $\int_x^{\infty} z^{-2}\log^{-2\alpha} z dz \xrightarrow{\text{large } x} x^{-1}\log^{-2\alpha} x$. Plugging in Equation 12, we find that the RHS of Equation 13 approaches $\frac{n}{\log\log n}$ at large $n$, and is thus $o(n)$. This implies benignness as in the proof of clause (a).

***Clause (b):*** *if* $\delta = 0$ *and* $\lambda_i = i^{-\alpha}$ *for some* $\alpha > 1$, *then* $\lim_{n\to\infty} \mathcal{E}_0 = \alpha$.

The desired limit follows from direct computation upon replacing the sum over eigenvalues in the definition of $\kappa$ with an integral. To make the result rigorous, we shall simply bound this sum between two integrals.

The constant $\kappa$ satisfies

$$\sum_{i=1}^{\infty} \frac{i^{-\alpha}}{i^{-\alpha}+\kappa} = \sum_{i\geq 1}^{\infty} \frac{1}{1+\kappa i^{\alpha}} = n. \tag{14}$$

Noting that the summand decreases monotonically with $i$, we find that

$$n = \sum_{i=1}^{\infty} \frac{1}{1+\kappa i^{\alpha}} \geq \int_{1}^{\infty} \frac{di}{1+\kappa i^{\alpha}} \geq \int_{0}^{\infty} \frac{di}{1+\kappa i^{\alpha}} - 1 \tag{15}$$

and

$$n = \sum_{i=1}^{\infty} \frac{1}{1+\kappa i^{\alpha}} \leq \int_{0}^{\infty} \frac{di}{1+\kappa i^{\alpha}}. \tag{16}$$

It follows that $\kappa_- \leq \kappa \leq \kappa_+$, where $\kappa_-$ and $\kappa_+$ satisfy

$$\int_{0}^{\infty} \frac{di}{1+\kappa_- i^{\alpha}} = n+1, \tag{17}$$

$$\int_{0}^{\infty} \frac{di}{1+\kappa_+ i^{\alpha}} = n. \tag{18}$$

Using the fact that $\int_0^\infty (1+x^\alpha)^{-1} dx = \alpha^{-1}\pi \csc(\pi/\alpha)$ when $\alpha > 1$, we find that

$$\kappa_- = \left[\frac{\pi}{\alpha}\csc(\pi/\alpha)\right]^{\alpha} n^{-\alpha}, \tag{19}$$

$$\kappa_+ = \left[\frac{\pi}{\alpha}\csc(\pi/\alpha)\right]^{\alpha} (n+1)^{-\alpha}. \tag{20}$$

These bounds converge as $n \to \infty$, and thus asymptotically $\kappa \to \left[\frac{\pi}{\alpha}\csc(\pi/\alpha)\right]^{\alpha} n^{-\alpha}$.

A similar argument using the integral $\int_0^\infty (1+x^\alpha)^{-2} dx = \alpha^{-2}(\alpha-1)\pi\csc(\pi/\alpha)$ yields that

$$\sum_{i=1}^{\infty} \mathcal{L}_i^2 \approx \int_{0}^{\infty} \frac{di}{(1+\kappa i^{\alpha})^2} \to \kappa^{-1/\alpha} \frac{\alpha-1}{\alpha^2}\pi\csc(\pi/\alpha) \to n\frac{\alpha-1}{\alpha}, \tag{21}$$

where in the last step we have inserted our asymptotic expression for $\kappa$. We now find that $\mathcal{E}_0 = \frac{n}{n-\sum_i \mathcal{L}_i^2} \to \alpha$.

***Clause (c):*** *if $\delta = 0$ and $\frac{\lambda_i}{\lambda_{i+1}} \geq \frac{i^{-\log i}}{(i+1)^{-\log(i+1)}}$ for all $i$, then* $\lim_{n\to\infty} \mathcal{E}_0 = \infty$.

We begin by observing that

$$\frac{1}{\mathcal{E}_0} = \frac{1}{n}\left(n - \sum_j \mathcal{L}_j\right) = \frac{1}{n}\sum_j \mathcal{L}_j(1-\mathcal{L}_j) \leq \frac{1}{n}\sum_{i\leq n}(1-\mathcal{L}_i) + \frac{1}{n}\sum_{i>n}\mathcal{L}_i. \tag{22}$$

We shall show that both terms on the RHS of Equation 22 approach zero as $n \to \infty$, and thus $\mathcal{E}_0$ diverges.

Considering the first term first, we note that

$$\frac{1}{n}\sum_{i\leq n}(1-\mathcal{L}_i) = \frac{1}{n}\sum_{i\leq n}\frac{\kappa}{\lambda_i+\kappa} \overset{(1)}{\leq} \gamma + \frac{\kappa}{n}\sum_{i\leq n(1-\gamma)}\frac{1}{\lambda_i} \tag{23}$$

$$\leq \gamma + \frac{\kappa}{\lambda_{n(1-\gamma)}} \overset{(2)}{\leq} \gamma + \frac{1}{n\gamma}\sum_{i\geq n(1-\gamma)} \frac{i^{-\log i}}{(n(1-\gamma))^{-\log(n(1-\gamma))}}, \tag{24}$$

where in step (1) we have ignored the last $n\gamma$ terms of the sum, and also used the fact that $\kappa/(\lambda_i+\kappa) \leq \kappa/\lambda_i$, and in step (2) we have used Equation 5 to upper-bound $\kappa$, with $\gamma$ a constant parameter we will later take to be small. Fixing $\gamma$ and taking $n \to \infty$, the final RHS of Equation 23 approaches $\gamma$. By making $\gamma$ arbitrarily small, we can make this upper bound approach zero.

Now examining the second term in Equation 22, we observe that

$$\frac{1}{n}\sum_{i>n}\mathcal{L}_i = \frac{1}{n}\sum_{i>n}\frac{\lambda_i}{\lambda_i+\kappa} \overset{(1)}{\leq} \gamma + \frac{1}{n\kappa}\sum_{i>n(1+\gamma)}\lambda_i$$
$$\overset{(2)}{\leq} \gamma + \frac{1}{n\gamma\lambda_{n(1+\gamma)}}\sum_{i>n(1+\gamma)}\lambda_i \leq \gamma + \frac{1}{n\gamma}\sum_{i\geq n(1+\gamma)}\frac{i^{-\log i}}{(n(1+\gamma))^{-\log(n(1+\gamma))}}, \quad (25)$$

where in step (1) we have again "given up" on $n\gamma$ terms of the sum and in step (2) we have used Equation 6 to lower-bound $\kappa$. Using the same argument as above, the final RHS of Equation 25 approaches $\gamma$, which can be made arbitrarily small.

Combining subresults and looking again at Equation 22, we find that $\lim_{n\to\infty}\mathcal{E}_0^{-1}$ can be given an arbitrarily small upper-bound, and thus $\lim_{n\to\infty}\mathcal{E}_0 = \infty$, which proves the clause. □

**Remarks on proofs and proof techniques**. Clause (a) shows that fitting is benign if $\delta > 0$, but it is not hard to show that fitting is also benign if $\delta = 0$ and instead $\delta' \equiv \lim_{j\to\infty}\sum_{i\geq j}\lambda_i > 0$. This odd tailsum $\delta'$ can be shown to act like an effective ridge parameter in Equation 3, but with $\delta = 0$ the resulting kernel interpolates the data. One way to add such an effective ridge to a kernel $K$ is to replace it with $K'(x_1, x_2) \equiv K(x_1, x_2) + \delta' \mathbb{1}_{x_1=x_2}$, which (assuming train and test sets are disjoint) simply adds a ridge parameter $\delta'$ to Equation 1 defining KR. The fact that many small eigenvalues can act like a ridge parameter is proven by and used in the derivations of Simon et al. [2021].

When proving clause (c), we used a ratio condition to capture the notion of super-powerlaw decay. We found that this ratio condition easier to work with than the weaker requirement that $\lambda_i = \mathcal{O}(i^{-\log i})$. We note that there was nothing special in the choice of $i^{-\log i}$, and any slower super-powerlaw decay, like $i^{-\log\log i}$, would have also sufficed.

## B Powerlaw and Laplace kernel experiments

In Section 3 of the main text, we show via Theorem 3.1 that KR with a kernel with a powerlaw spectrum with exponent $\alpha$ overfits target noise by a factor $(\alpha - 1)$ as $n \to \infty$, and we conclude that Laplace kernels, which are known to have powerlaw spectral tails, ought to obey this rule[12]. Here we provide experimental evidence for both these claims.

We start with KR with a powerlaw spectrum. No simple kernel + domain pair gives an exact powerlaw spectrum $\lambda_i = i^{-\alpha}$, so we perform a synthetic experiment with Gaussian random eigenfunctions, which is essentially linear regression with random features as in the setting of Bartlett et al. [2020]. Letting $n$ denote the number of train samples, $n'$ denote the number of test samples, and $M \gg n$ denote the number of eigenmodes (a.k.a. features), we first sample train and test feature matrices $\Phi \in \mathbb{R}^{M\times n}$ and $\Phi' \in \mathbb{R}^{M\times n'}$ with entries drawn i.i.d. from $\mathcal{N}(0,1)$. We then construct the train-train and test-train kernels as $K_{\text{tr-tr}} = \Phi^T\Lambda\Phi$ and $K_{\text{te-tr}} = \Phi'^T\Lambda\Phi$, respectively, where $\Lambda \equiv \text{diag}(\lambda_1, \cdots, \lambda_M)$ with $\lambda_i = i^{-\alpha}$.

The train and test targets are given by $\mathcal{Y}_{\text{tr}} = \Phi^T\mathbf{v} + \eta_{\text{tr}}$ and $\mathcal{Y}_{\text{te}} = \Phi'^T\mathbf{v} + \eta_{\text{te}}$, respectively, where $\mathbf{v} \in \mathbb{R}^M$ is a vector of target eigencoefficients and the noise vectors are sampled $\eta_{\text{tr}} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n)$ and $\eta_{\text{tr}} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_{n'})$. The predicted test labels are computed via KR as $\widehat{\mathcal{Y}}_{\text{te}} = K_{\text{te-tr}}K_{\text{tr-tr}}^{-1}$ and the MSE is then $\frac{1}{n'}|\mathcal{Y}_{\text{te}} - \widehat{\mathcal{Y}}_{\text{te}}|^2$.

We run experiments varying $n$ and using $n' = 3000$, $M = 10^4$, $\mathbf{v}_i \sim i^{-2}$ normalized so $\sum_{i=1}^M \mathbf{v}_i^2 = 10$, $\sigma^2 = 1$, and varying $\alpha$. The results, shown in Figure 5, confirm that, as $n$ grows, MSE approaches $\alpha\sigma^2$ in accordance with Theorem 3.1b.

[12] As a reminder, the asymptotic MSE was $\alpha\sigma^2$, while the Bayes-optimal value was $\sigma^2$, so the excess risk is $(\alpha-1)\sigma^2$ and we say the noise is overfit by a factor of $(\alpha - 1)$.

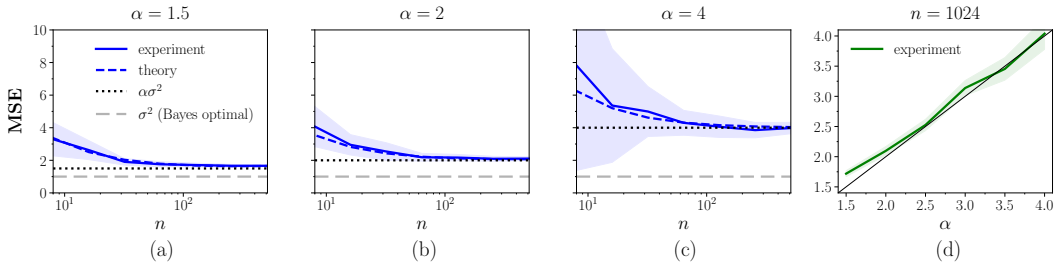



Figure 5: **As $n$ grows, the MSE of KR with Gaussian eigenfunctions and powerlaw kernel eigenspectra with exponent $\alpha$ approaches $\alpha\sigma^2$. (a-c)**: learning curves with different $\alpha$. **(d)**: test MSE at $n = 1024$ for varying $\alpha$, with the identity function shown by the solid line.

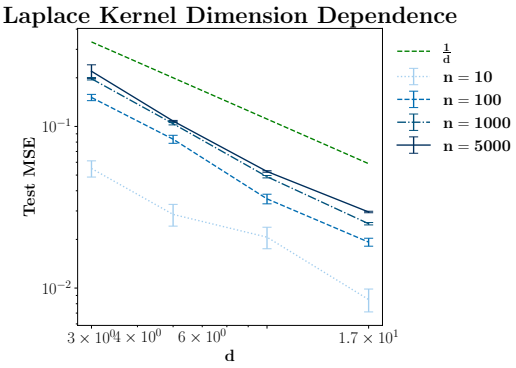



Figure 6: **As $n$ grows, the excess MSE of Laplace KR trained on noisy data on $\mathcal{S}^d$ decays like $1/d$.**

We now study proper KR with a Laplace kernel and data sampled i.i.d. from the hypersphere $\mathcal{S}^d \equiv \{x \in \mathbb{R}^{d+1} | x^2 = 1\}$. We run Laplace KR with varying $d$ and increasing $n$, using a kernel bandwidth of 1, pure-noise train targets of $\mathcal{N}(0, 1)$, and test samples with noiseless, uniformly-zero test targets so the resulting MSE is simply the excess risk. As shown in Figure 6, we find that, for large $n$, excess MSE appears to decay as $\Theta(1/d)$ as expected from the kernel eigendecay[13].

# C Experimental Details

## C.1 Datasets

Our experiments are performed using the following real and synthetic datasets:

- **The d-sphere**: Data is sampled uniformly from $\mathcal{S}^d$ with noisy Gaussian targets. We use the convention that $d$ is the manifold dimension, and so $S^d$ is the hypersphere embedded in $\mathbb{R}^{d+1}$.
- **MNIST**. [LeCun, 1998, LeCun and Cortes, 2010]
- **Binary MNIST**. We binarize MNIST by classifying even vs. odd digits.
- **CIFAR-10**. [Krizhevsky et al., 2009]
- **Binary CIFAR-10**: We binarize CIFAR-10, forming two classes of animals (bird, cat, deer, dog) and vehicles (airplane, automobile, ship, truck).[14]
- **0/1 SVHN**. We binarize SVHN [Netzer et al., 2011], selecting the digits zero and one. We use the "train" and "extra" datasets for train data from the `torchvision` library.

[13] We note that it is unclear from this experiment whether excess MSE approaches $1/d$ *exactly* as $n \to \infty$ or instead converges to roughly $c/d$ for some constant $c < 1$.

[14] We drop the "frog" and "horse" classes to create a balanced dataset.

- **CINIC-10**. [Darlow et al., 2018] 10-class image dataset with samples sourced from CIFAR-10 and Tiny ImageNet.

We obtained the MNIST, CIFAR-10, and SVHN datasets from the `torchvision` library.[15]

In all experiments with subsampled train datasets, we uniformly sample over all training data available, and experimental results are averaged over multiple samples (and random seeds). All results are computed across the entire test set.

### C.2 Experimental Setups

Unless otherwise stated, models are trained with no weight decay, dropout, or other regularization techniques, as is done in Zhang et al. [2017]. This encourages interpolation of training datasets even at high label noise levels. As is standard, we perform channel-wise normalization of image datasets.

During training of Binary CIFAR-10, we additionally employ random horizontal flips. On MNIST, CIFAR-10 and SVHN, we do not use flips or crops. As a result of this, and of our avoidance of regularization techniques, our models underperform current state-of-the-art methods, as is to be expected when intending to improve the state of understanding, not the state of the art.

We train the following models:

**Kernel Regression**: For synthetic data on $\mathcal{S}^d$ we directly solve the kernel regression problem by inverting the data kernel matrix. On MNIST, we create flattened vectors out of image data and train Gaussian and Laplacian kernels with stochastic gradient descent (SGD) in function space using EigenPro [Ma and Belkin, 2017]. EigenPro automatically computes an optimal batch size given the number of training data points and available memory on the GPU being used.[16]

**Neural Networks**: We train multi-layer perceptrons (MLPs) and Wide ResNets (WRNs), and VGG networks [Simonyan and Zisserman, 2015] using SGD with batch size 128, initial learning rate of 0.1, and momentum of 0.9. For the MLPs on synthetic data and WRNs on SVHN we employ a learning rate decay factor of 0.1; for WRNs on CIFAR-10 we use a decay factor of 0.2. Per-experiment details on specific learning rate schedules are given in Appendix C.3.

Initial parameters for synthetic experiments are chosen by experimentation and largely guided by commonly used settings from non-synthetic cases. For WRNs, the learning rate decay factors, momentum, and batch size choices for CIFAR-10 and SVHN are chosen based on Zagoruyko and Komodakis [2016]. Wide ResNet model code is sourced from a publicly available Git repository.[17] MLP model code and training/testing scripts are written in-house. Neural networks and EigenPro kernels are trained using PyTorch[18].

### C.3 Experiment-Specific Details

**MLP and Nearest Neighbor Noise Profiles on Binary MNIST (Figure 3)** We train $3 \times 1024$ MLPs, 1-NN, and $k$-NN ($k \sim \log n$) models on Binary MNIST. Results are averaged over five independent runs with mean and standard error bars reported. Nearest Neighbor models use default settings from the `scikit-learn` library.[19] MLPs are trained using Adam, a learning rate of $1 \times 10^{-3}$, momentum of 0.9, batch size of 256. The learning rate is decayed by a factor of 0.1 at epochs 60 and 90. No learning rate warmup is used. All models are trained until the train loss reaches $1 \times 10^{-4}$ and achieve 100% training accuracy.

**Asymptotics of Kernel Regression on $d$-Sphere (Figure 4)** In this experiment, provided in Figure 4, we sample data, $\{x_i\}_i$, uniformly on $\mathcal{S}^d$ and train with pure noise target labels $y_i \sim \mathcal{N}(0, 1)$. Test MSE is computed on clean labels, $y_i = 0$. In experiments with a ridge, we use $\delta = 0.1$. The kernel regression problem is solved by directly inverting the training data-data kernel matrix against the

[15] `https://pytorch.org/vision/stable/datasets.html`

[16] `https://github.com/EigenPro/EigenPro-pytorch`

[17] `https://github.com/meliketoy/wide-resnet.pytorch/blob/master/networks/wide_resnet.py`

[18] `https://pytorch.org/`

[19] `https://scikit-learn.org/stable/`

noisy labels. For this plot the lines represent the median of 100 independent runs and error regions are given for 25% and 75% quantiles.

**Kernel Noise Profiles on MNIST (Figure 7)** Noise profiles for ridgeless Laplacian and Guassian kernels are provided for 10-class classification on MNIST. All models reach 100% training accuracy and $\leq 10^{-4}$ train MSE. Results are averaged over five independent runs.

**MLP Noise Profiles on 10-Sphere (Figure 8b)** In this experiment, we take $f^*(x) = 1$ and inject label noise in the form of randomly flipping a fixed proportion, $p \in [0, 0.5]$, of training labels to $-1$. The Bayes optimal classifier for $p < 0.5$ is $\mu(x) = 1$. The goal of this experiment is to show that even in the simplest case of learning a constant function, interpolating neural networks suffer from noisy training data. We present noise profiles for classification error as a function of training label flip probability in Figure 8b.

The MLP is a $3 \times 1024$ network trained using SGD with initial learning rate of 0.1, momentum of 0.9, and batch size of 128. For $n < 120000$ we cut the learning rate by a factor of 0.1 at epoch 150 and again at epoch 350. For $n = 120000, 360000$ we cut the learning rate by a factor of 0.1 at epoch 500 and again at epoch 750.

In all experiments, the stopping criteria is a train MSE loss $\leq 1e^{-4}$, regardless of where the learning rate schedule has reached by that point. Controlling for train loss stopping point allows us to meaningfully compare models that have achieved the "same level of overfitting" and not introduce confounding factors due to late stage training effects. We average results over five independent runs.

**WRN Noise Profiles on (Binary) CIFAR-10 and SVHN (Figures 2, 8a, 9)** In all experiments, we train the Wide ResNet $28 \times 10$ for 60 total epochs using SGD with initial learning rate of 0.1, momentum of 0.9, and batch size of 128. We cut the learning rate by a factor of 0.2 in CIFAR-10 experiments and 0.1 in SVHN experiments. The learning rate in both settings is cut once at epoch 30, and again at epoch 40.

Classification label noise is injected for each point $\{(x_i, y_i)\}_i$ by uniformly re-sampling labels from alternative class labels, excluding the ground truth label. We resample labels in this way for a fraction of the dataset, $p$. For 10 class and binary classification we vary $p \in [0.0, 0.9]$ and $p \in [0.0, 0.5]$, respectively. For example, $p = 0.9$ indicates that we train with exactly $90\%$ label noise.

Test classification error is computed on the clean test set. For Binary CIFAR-10, CIFAR-10, and 0/1 SVHN we average results over three independent runs and report mean and standard error bars. On 0/1 SVHN, for each noise level and train set size, $n$, we perform 2-fold cross-validation to select the early-stop epoch which maximizes validation classification error averaged over both folds. We additionally do not consider the first ten epochs for early-stopping, during the learning rate warmup phase. We create the early-stopped noise profile from the classification test error attained at each of the early-stopped epoch choices.

**MLP Error vs. Time (Figure 11)** We train a $4 \times 512$ ReLU MLP trainsets of various sizes with $\{x_i\}$, sampled uniformly on $\mathcal{S}^4$. Training labels are given by $y_i = 1 + \mathcal{N}(0, 1)$, a distribution for which the Bayes-optimal MSE is 1. To reduce statistical error, we compute test MSE with clean labels $y_i = 1$ and simply add 1 by hand to account for the noise. We average over five trials and plot against $t \equiv$ [learning rate] $\times$ [epoch number]. We train with full batch gradient descent using JAX and the neural-tangents library.[20]

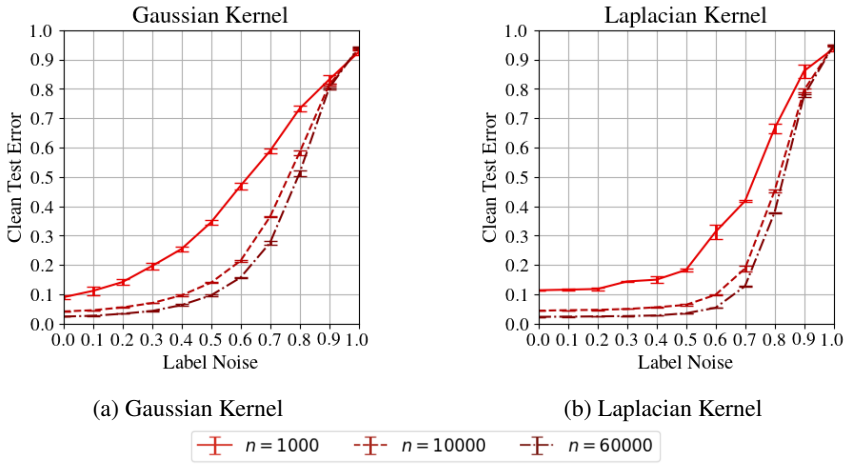
**VGG Noise Profile on CINIC-10 (Figure 10)** We train a standard VGG-19 model with batch norm on the CINIC-10 dataset. We use SGD with initial learning rate 0.1 and momentum 0.9. We train for 200 epochs and decay the learning rate by a factor of 0.1 at epoch 60 and epoch 120.

### C.4 Compute Details

Each neural network experiment and kernel experiment using EigenPro was run on a single NVIDIA V100 GPU. Experiments were performed in parallel using Expanse GPU nodes on XSEDE. [Towns

[20] https://github.com/google/neural-tangents

et al., 2014] The final experiments for this paper took roughly 250 GPU-hours, and the experimentation phase of the project took roughly 1500 GPU-hours.




(a) Gaussian Kernel (b) Laplacian Kernel

Figure 7: We show noise profiles Gaussian and Laplacian kernels trained with label noise on 10-class MNIST for classification. We plot test classification error on the clean test set (no label noise) vs. training label noise. Overfitting is considered when the train MSE is $\leq 10^{-4}$. All models achieve 0 classification error on the training set.

# D Additional Experimental Results

## D.1 Kernel Noise Profiles on MNIST

We provide additional experimental results here for kernel noise profiles trained on MNIST data, as described in Appendix C.3.

In Figure 7 we show noise profiles for Gaussian and Laplacian kernels trained on MNIST data at varying values of training data size, $n$. We immediately see from the plots that the Laplacian kernel is more tempered than the Gaussian kernel.

Note that while the Gaussian kernel does not look catastrophic, the input dimension of MNIST is larger than all experiments on $\mathcal{S}^{d-1}$ shown in Figure 4. In this case, we would need a significantly larger number of training samples to observe the Gaussian kernel entering the catastrophic regime. Synthetic experiments on $\mathcal{S}^{1000}$ (not shown) up to 1 million training samples show that the Gaussian kernel still does not enter the catastrophic regime, therefore with MNIST only containing 60k samples we see a more tempered noise profile.

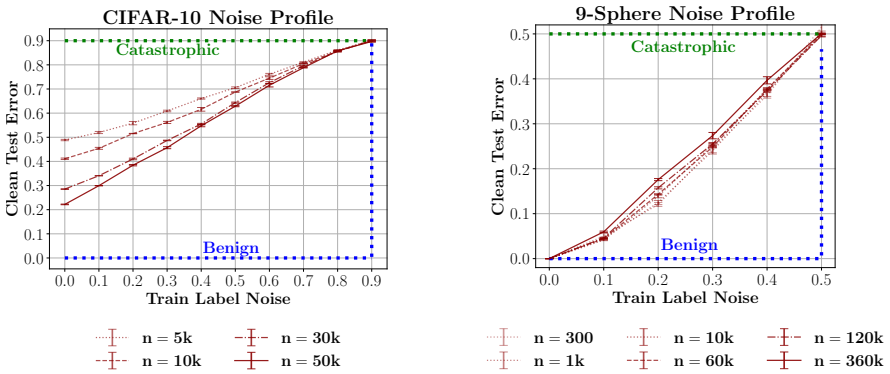
## D.2 Additional DNN Noise Profiles

Here we report additional DNN noise profiles for ResNets on CIFAR-10 (Figure 8a), MLPs on synthetic data on $\mathcal{S}^9$ (Figure 8b), and ResNets on binarized SVHN both optimally-early-stopped (Figure 9a) and trained to interpolation (Figure 9b).

## D.3 VGG Noise Profile

We additionally explore noise profiles for overfit VGG networks on the CINIC-10 dataset. We do so in order to show that both residual and non-residual convolutional networks, on differing image datasets, maintain a tempered noise profile. In Figure 10 we plot a 19-layer VGG network with batch norm overfit to the ten class CINIC-10 dataset.

On this dataset we were only able to run one experiment due to computational limitations, however in other experiments with deep neural networks we note low variance over the test error of DNNs at




(a) WideResNet trained on CIFAR-10. (b) MLP trained on the constant function.

Figure 8: **Noise profiles for interpolating DNNs on CIFAR-10 and synthetic classification tasks.** In both settings, the noise profiles asymptotically behave as *tempered* overfitting: neither catastrophic nor benign.

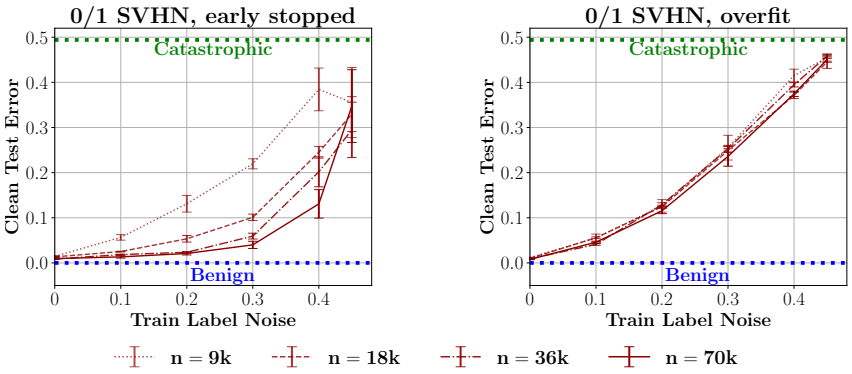



Figure 9: **Early-stopped DNNs exhibit fairly benign fitting, while DNNs trained to interpolation exhibit tempered overfitting.** We show noise profiles for ResNets trained on Binary SVHN (0/1 digit classification). **(a)** With early stopping, noise profiles approach zero test error as $n$ increases, even with finite label noise, indicating benign fitting. **(b)** When training to interpolation, noise profiles converge to a limiting, roughly linear curve as $n$ increases, indicating tempered fitting.

interpolation. Therefore, we expect the same to hold true over multiple runs of the VGG network on CINIC-10.

## E Regimes of fitting throughout DNN training

The discussion in the main text suggests that as a single DNN is trained, it exhibits benign fitting early in training (when it has not fit the noise in its train set), and then transitions to tempered overfitting late in training (as it eventually fits the noise). In Figure 11, we show a simple experiment that illustrates these time dynamics. We train a ReLU MLP on the following synthetic regression task (similar to Figure 8b): The inputs $x$ are sampled from the unit sphere $\mathcal{S}^4$, and the targets are $\mathcal{N}(1, 1)$. That is, we are simply trying to learn the constant function $f^*(x) = 1$ on the unit sphere, with Gaussian observation noise. We then plot the test MSE as a function of the number of optimization steps taken.

From Figure 11, we see that for all train sizes $n$, the network quickly reaches nearly Bayes-optimal test performance early in training. In this initial fitting phase, train and test MSE decrease together: the network is fitting the ground-truth function but has not yet fit the noise. Once the train MSE drops below the Bayes risk, however, the dynamics change. At this point (around time $t = 4$ in Figure 11), the network must start to fit noise in the train set, since the train loss drops below the optimal possible test loss. As training continues, we see the train loss decrease, but the test loss

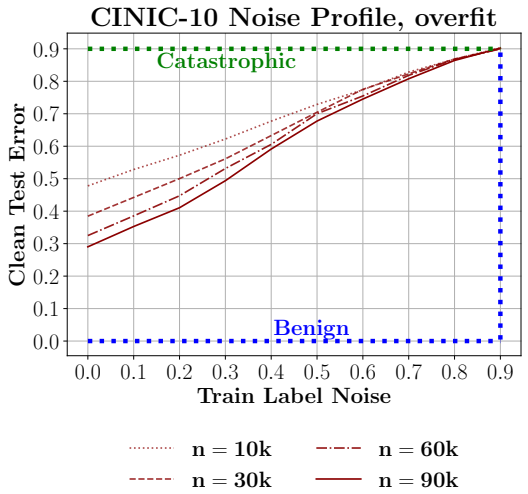



Figure 10: VGG-19 with batch norm overfit to the CINIC-10 dataset. We plot test classification error on the clean test set vs. training set label flip probability.

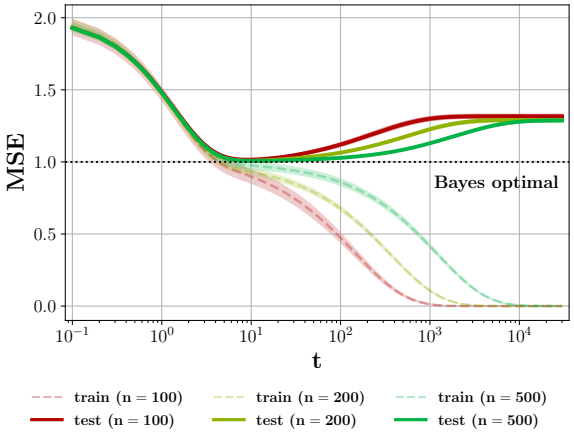



Figure 11: **MLPs trained on noisy data fit *benignly* at early times and exhibit *tempered overfitting* at late times.** Curves show train and test MSE for a $4 \times 500$ ReLU MLP with target labels $y_i \sim \mathcal{N}(1, 1)$ and varying trainset size $n$. Benign fitting, with near-Bayes-optimal test MSE, begins around $t = 3$, ending as the network begins to fit the noise in the training set. Tempered overfitting is reflected in the suboptimal-but-finite asymptotic test MSE as $t$ becomes large.

*increase* for the remainder of training: fitting noise in the train set hurts test performance. Finally, in the limit of large time $t$, the network converges to a test loss that is tempered: above Bayes-optimal, but asymptotically constant as a function of samples $n$.