Journal of Machine Learning Research 24 (2023) 1-59 Submitted 9/21; Published 10/23

Revisiting minimum description length complexity in overparameterized models

Raaz Dwivedi⋆ DWIVEDI@CORNELL.EDU
Department of Operations Research & Information Engineering
Cornell Tech, Cornell University
New York City, NY

Chandan Singh⋆ CHANSINGH@MICROSOFT.COM
Microsoft Research
Seattle, WA

Bin Yu BINYU@BERKELEY.EDU
Department of Statistics, and Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA

Martin Wainwright WAINWRIGWORK@GMAIL.COM
Department of Electrical Engineering and Computer Sciences, and Mathematics
Massachusetts Institute of Technology
Cambridge, MA
⋆ = *equal contribution.*

Editor: Jean-Philippe Vert

Abstract

Complexity is a fundamental concept underlying statistical learning theory that aims to inform generalization performance. Parameter count, while successful in low-dimensional settings, is not well-justified for overparameterized settings when the number of parameters is more than the number of training samples. We revisit complexity measures based on Rissanen's principle of minimum description length (MDL) and define a novel MDL-based complexity (MDL-COMP) that remains valid for overparameterized models. MDL-COMP is defined via an optimality criterion over the encodings induced by a good Ridge estimator class. We provide an extensive theoretical characterization of MDL-COMP for linear models and kernel methods and show that it is *not* just a function of parameter count, but rather a function of the singular values of the design or the kernel matrix and the signal-to-noise ratio. For a linear model with n observations, d parameters, and i.i.d. Gaussian predictors, MDL-COMP scales linearly with d when $d < n$, but the scaling is exponentially smaller—$\log d$ for $d > n$. For kernel methods, we show that MDL-COMP informs minimax in-sample error, and can decrease as the dimensionality of the input increases. We also prove that MDL-COMP upper bounds the in-sample mean squared error (MSE). Via an array of simulations and real-data experiments, we show that a data-driven Prac-MDL-COMP informs hyper-parameter tuning for optimizing test MSE with ridge regression in limited data settings, sometimes improving upon cross-validation and (always) saving computational costs. Finally, our findings also suggest that the recently observed double decent phenomenons in overparameterized models might be a consequence of the choice of non-ideal estimators.

Keywords: Complexity, minimum description length, high-dimensional models, ridge regression, kernel regression

1. Introduction

Occam's razor and the bias-variance tradeoff have long provided guidance for model selection in statistics and machine learning. Given a dataset, these principles recommend selecting a model that balances between (a) fitting the data well, and (b) having relatively low complexity. Roughly speaking, in the low-dimensional regime, one typically observed the following tradeoff: On the one hand, a model whose complexity is too low incurs *high bias*, i.e., it does not fit the training data well (under-fitting); on the other hand, an overly complex model memorizes the training data, suffering from *high variance*, i.e. poor performance on unforeseen data (over-fitting).

There are numerous characterizations of complexity in the statistical machine learning literature that are commonly used to perform model selection, and to establish bounds on prediction error. These include Akaike information criterion (Akaike, 1974), Mallow's C_p (Mallows, 1973), Bayesian information criterion (Schwarz, 1978), Vapnik-Chervonenkis dimension (Vapnik and Chervonenkis, 2015), and Rademacher complexity (Bartlett and Mendelson, 2002; Anthony and Bartlett, 2009; Van De Geer, 2006). Intuitively, such measures reflect the effective number of parameters used to fit the model. In the classical regime for linear regression with d features, and n training data points, when $d < n$, and the design matrices are well-conditioned, all of these measures reduce to simple parameter counting for linear models. Another common proxy for model complexity is the degrees of freedom (Efron, 1986; Buja et al., 1989; Efron, 2004), along with extensions such as effective or generalized degrees of freedom for more complex models (Meyer and Woodroofe, 2000; Shen and Ye, 2002; Efron, 2004; Hastie et al., 2005; Zhang et al., 2012), and high-dimensional sparse regression (Zou et al., 2007; Tibshirani and Taylor, 2012; Hastie, 2017). As a special case, in an unstructured linear regression problem based on n samples with d features (with $d < n$), the degree of freedom is equal to d. However, recent work (Kaufman and Rosset, 2014; Janson et al., 2015; Tibshirani, 2015) has shown that when moving to the over-parameterized setting ($d > n$), effective degrees of freedom may be a poor measure of model complexity. In some past work, the variance of the estimator itself has sometimes been used as a measure of complexity (e.g., in L^2-boosting (Bühlmann and Yu, 2003)). However, such a choice may be misinformed when the bias term is dominant.

Overall, the choice of parameter count as a complexity measure for linear models is rigorously justified when the data is low-dimensional ($d < n$), and the design matrix is well-conditioned (so that all d directions contribute in roughly equal measure). Indeed, under these conditions, the OLS estimate has good performance when $d \ll n$, and its test error increases proportionally with d in this regime. However, using d as the complexity measure when $d > n$ remains unjustified even for linear models, and for low-dimensional linear models when the design matrix is not well-conditioned; and in high-dimensional models, the design matrix by definition is ill-conditioned since it does not even have full rank.

More recently, a line of work has derived generalization error bounds for deep neural networks (Neyshabur et al., 2015; Bartlett et al., 2017; Neyshabur et al., 2017; Golowich et al., 2017; Li et al., 2018b; Neyshabur et al., 2019) based on Rademacher-like complexity

notions. However, at least thus far, such bounds remain too loose to inform practical performance (Arora et al., 2018). Moreover, there is growing evidence that heavily overparameterized models once trained are often not complex, due to the implicit regularization induced by model architecture, optimization methods including initialization, and training datasets (Nakkiran et al., 2019; Neyshabur et al., 2014; Arora et al., 2019; Neyshabur et al., 2019). There has been a series of recent work investigating generalization performance of overparameterized linear models and kernel methods as they can be seen as as tractable settings for providing theoretical insight into the behavior of (overparameterized) deep neural networks. See, e.g., Belkin et al. (2018); Jacot et al. (2018); Du et al. (2018); Allen-Zhu et al. (2018); Hastie et al. (2019); Bartlett et al. (2020); Tsigler and Bartlett (2020) and the references therein.

The starting point for this work is to seek a valid complexity measure for regression tasks with overparameterized linear models, that can be easily extended to kernel methods, and then to use this complexity measure for tuning regularization parameter. To do so, we build on the optimality principle put forth in the algorithmic complexity of Kolmogorov, Chaitin, and Solomonoff (Kolmogorov, 1963, 1968; Li and Vitányi, 2008) and the principle of minimum description length (MDL) of Rissanen (Rissanen, 1986; Barron et al., 1998; Hansen and Yu, 2001; Grünwald, 2007). For linear models, the known MDL complexity measures also scale roughly linearly with dimension d, or they can be infinite (see Sec. 2.3)

Our contributions: We define a new complexity measure MDL-COMP that corresponds to the minimum excess bits required to transmit the data when constructing an optimal code for a given dataset via a family of models based on the ridge estimators. Within this framework, we undertake a detailed study of high-dimensional linear models and kernel methods.

We show that MDL–COMP has a wide range of behavior depending on the design matrix (or the kernel matrix). For linear models with d features, and n samples, it usually scales like d/n for $d < n$, and grows very slowly—logarithmic in d—for $d > n$ (see Figs. 1 and 2). We establish that MDL–COMP provides an upper bound for in-sample MSE (Theorem 2), and that it satisfies certain minimax optimality criterion (Theorem 5). For kernel methods, we show that for kernels in Gaussian and Sobolev spaces, MDL-COMP can inform the minimax in-sample generalization (Theorem 3 and Corollary 1). Interestingly, for neural tangent kernels (NTKs), we find that MDL-COMP itself can sometimes reduce when the dimensionality of the input increases.

Next, to evaluate the practical usefulness of MDL–COMP, we consider a data-driven form of MDL–COMP-inspired hyperparameter selection, which provides competitive performance with cross-validation for ridge regression (in terms of test MSE), especially in limited data settings in several simulations and real-data experiments. Moreover, this criterion can provide computational savings especially while training overparameterized models in contrast to the vanilla K-fold cross-validation (since computation is only required for a single fold). Finally, we also highlight some insights that our findings provide for the recently observed *double descent* phenomenon on test error of overparameterized models.

Organization: We start with a background on MDL and setting the notation in Sec. 2, followed by the definitions of complexity central to this work in Sec. 3. We then provide our main results and their consequences in Sec. 4, and present several numerical experiments

in Sec. 5. We conclude with a discussion and directions for future work in Sec. 6, where we also discuss the consequences of our results for double-descent. Proofs of all results and additional experiments are provided in the appendix.

2. Background on the principle of minimum description length

In order to define a complexity measure in a principled manner, we build upon Rissanen's principle of minimum description length (MDL). It has its intellectual roots in Kolmogorov's theory of complexity. Both approaches are based on an optimality requirement: namely, the shortest length program that outputs a given sequence on a universal Turing machine for Kolmogorov complexity, and the shortest (probability distribution-based) codelength for the given data for MDL. Indeed, Rissannen generalized Shannon's coding theory to universal coding and used probability distributions to define a universal encoding for a dataset and then used the codelength as an approximate measure of Kolmogorov's complexity.

2.1 Basic principle

From the perspective of minimum description length, a model or a probability distribution for data is equivalent to a (prefix) code; one prefers a code that exploits redundancy in the data to compress it into as few bits as possible; see (Rissanen, 1986; Barron et al., 1998; Grünwald, 2007). Since its introduction, MDL has been used in many tasks including density estimation (Zhang, 2006), time-series analysis (Tanaka et al., 2005), model selection (Barron et al., 1998; Hansen and Yu, 2001; Miyaguchi and Yamanishi, 2018), and DNN training (Hinton and Van Camp, 1993; Schmidhuber, 1997; Li et al., 2018a; Blier and Ollivier, 2018). For readers unfamiliar with MDL, Sections 1 and 2 of the papers (Barron et al., 1998; Hansen and Yu, 2001) provide background on MDL.

In the MDL framework, any probability model is viewed as a type of coding, so that for example, fitting a Gaussian linear model is equivalent to using a Gaussian code for compressing the data. The goal is to select the code that provides the shortest description of data; in most settings, this translates into picking the model with the best fit to the data. More formally, suppose we are given a set of n observations $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$. Let $\mathbb{Q}$ refer to a probability distribution on the space $\mathcal{Y}^n$ (e.g., a subset of $\mathbb{R}^n$) where $\mathbf{y}$ takes values, and let $\mathcal{Q}$ denote a set of such distributions. Given a probability distribution $\mathbb{Q}$, we can associate an *information-theoretic* prefix code for the space $\mathcal{Y}^n$, wherein for any observation $\mathbf{y}$ we need to use $\log(1/\mathbb{Q}(\mathbf{y}))$ number of bits to *describe* $\mathbf{y}$. The MDL principle dictates choosing the code $\mathbb{Q}$ associated with the shortest possible description length—namely, a code achieving the minimum $\min_{\mathbb{Q}\in\mathcal{Q}} \log(1/\mathbb{Q}(\mathbf{y}))$. Note that when $\mathcal{Q}$ is simply a parametric family, the direct MDL principle reduces to the maximum likelihood principle. But the advantage of MDL comes from the fact that the set $\mathcal{Q}$ can be more complex, e.g. a nested union of parametric families, or a set of codes not indexed by the canonical parameter of interest. Furthermore, even without a generative modeling assumption, the notion of shortest description over an arbitrary set of codes $\mathcal{Q}$ continues to be well-defined. (For further discussion about MDL vs maximum likelihood, we refer the reader to Appendix A.6.)

2.2 Two-stage MDL

One of the earliest notions of MDL is two-stage MDL, often considered for doing model selection over a nested family of parametric model classes, where the dimensionality of the parameter varies across different parametric classes (Hansen and Yu, 2001). This version of MDL turns out to be equivalent to the Bayesian information criterion (BIC)—that is, it performs model selection based on a regularized maximum likelihood, where the regularization term is simply $\frac{d}{2}\log n$. Consequently, apart from the additional logarithmic term, the MDL complexity in this set-up simply reduces to parameter counting.

2.3 Normalized maximum likelihood

Many modern approaches to MDL are based on a form of universal coding known as *normalized maximum likelihood*, or NML for short (Shtar'kov, 1987). In this approach, the distribution $\mathbb{Q}$ is defined directly on the space $\mathcal{Y}^n$; at least in general, it is *not* explicitly parametrized by the parameter of interest. More concretely, given a family of codes $\mathcal{P}_\Theta = \{p(\cdot;\theta), \theta \in \Theta\}$, the NML code is defined as

$$q_{\mathrm{NML}}(\mathbf{y}) := \frac{\max_\theta p(\mathbf{y};\theta)}{\int_{\mathcal{Y}^n} \max_{\theta'} p(\mathbf{y}';\theta')d\mathbf{y}'}, \tag{1}$$

assuming that the integral in the denominator is finite. Shtar'kov (1987) established that this NML distribution (when defined) provides the best encoding for the family $\mathcal{P}_\Theta$ in a minimax sense. The log normalization constant

$$\log \int_{\mathcal{Y}^n} \max_{\theta'} p(\mathbf{y}';\theta')d\mathbf{y}' \tag{2}$$

associated with this code is referred to as the *NML or Shtarkov complexity.* Note that the normalization (1) ensures that $\mathbb{Q}_{\mathrm{NML}}$ is a valid code by making q_{NML} a valid density.[1] Such codes are called *universal codes*, since the codelength $\log(1/q_{\mathrm{NML}}(\mathbf{y}))$ is universally valid for any $\mathbf{y} \in \mathcal{Y}^n$.

Known results: Suppose that $\mathcal{P}_\Theta$ is a parametric class of dimension d. In this special case, under suitable regularity conditions, the NML complexity scales asymptotically as $\frac{1}{2}d\log n$, for fixed d with $n \to \infty$; consequently, it is asymptotically equivalent to the BIC complexity measure (Barron et al., 1998; Foster and Stine, 2004; Hansen and Yu, 2001) to the first order $\mathcal{O}(\log n)$ term. In recent work, Grünwald and Mehta (2019) further developed a framework to unify several complexities including Rademacher and NML, and derived excess risk bounds in terms of this unifying complexity measure, for several low-dimensional settings.

Challenges with NML complexity: In overparametrized settings (and even in several settings otherwise), the NML code suffers from the *infinity* problem, namely the normalization constant in equation (1) is infinite, and the NML distribution is not defined. A canonical solution is to truncate the observation space so as to make the integral finite. But with simple models like linear regression, such schemes provide volume of the truncated space as

[1] Thus, Kraft's inequality guarantees the existence of a code corresponding to it.

the complexity measure. For example, consider the codes corresponding to Gaussian linear model, where

$$p(\mathbf{y};\theta) \propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{X}\theta - \mathbf{y}\|_2^2), \tag{3}$$

and we treat the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and the scalar σ^2 known and fixed, and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^n$ denotes the observation. When $d > n$, and $\mathbf{X}$ has full row rank (n), it is straightforward to see that, $\max_\theta p(\mathbf{y};\theta) = \text{constant}$, which in turn implies that the NML complexity in equation (2) is infinite; and the NML code (1) is not defined (also, see Grünwald (2007, Example 11.1)). A canonical solution of this problem is truncation of the response space Barron et al. (1998), but in the setting above such a restriction leads to a trivial NML complexity measure that depends merely on the volume of the truncated space (and is independent of $\mathbf{X}, d$).

2.4 Luckiness normalized maximum likelihood

To deal with the infinity problem discussed above, another solution was proposed in recent works, namely, the luckiness normalized maximum likelihood (LNML) code (see Chapter 11 (Grünwald, 2007)). Given a class of codes $\mathcal{P}_\Theta$, and a luckiness function $p_{\text{luck}} : \Theta \to \mathbb{R}_+$ (not necessarily a code), one way to define the LNML code is as follows:

$$q_{\text{LNML}}(\mathbf{y}) := \frac{\max_\theta \left(p(\mathbf{y};\theta)\cdot p_{\text{luck}}(\theta)\right)}{\displaystyle\int_{\mathbf{z}\in\mathbb{R}^n} \max_{\theta'} \left(p(\mathbf{z};\theta')\cdot p_{\text{luck}}(\theta')\right)\, d\mathbf{z}}. \tag{4}$$

Once again, the normalization in equation (4) ensures that $\mathbb{Q}_{\text{LNML}}$ whenever well-defined is a universal code. One can now treat the log normalization constant of this code as a complexity measure, but such a definition would vary with the luckiness function chosen by the user. In the sequel, we provide a principled way to consider a family of such LNML codes and then derive a complexity measure using an optimality criterion over these codes. Theoretical investigations with LNML codes have not been extensively done in the prior work, and are central to the current work. It is worth noting that while NML can be seen as a generalization of the maximum likelihood principle, LNML can be interpreted as a generalization of regularized maximum likelihood principle (see Appendix A.6 for a detailed discussion about relationship betwee MDL and maximum likelihood).

It is useful to note the recent unified interpretation for the LNML codes equation (4) by Grünwald and Roos (2019). Given a luckiness function p_{luck}, (Grünwald and Roos, 2019) define an *MDL estimator based on* p_{luck} as follows:

$$\widehat{\theta}_{\text{luck}}(\mathbf{y}) := \arg\max_\theta p(\mathbf{y};\theta)\cdot p_{\text{luck}}(\theta). \tag{5}$$

Note that the term inside the arg on the RHS is the same term appearing in the numerator of equation (4). The p_{luck}-MDL estimator equation (5) is effectively a penalized maximum likelihood estimator, and coincides with the Bayes Maximum A Posteriori estimate (MAP) whenever p_{luck} is a probability density. Indeed, as we later discuss, here we investigate a family of MDL-estimators parameterized by positive semidefinite matrices $\mathbf{\Lambda}$ where for each estimator $p_{\text{luck}} = p_{\mathbf{\Lambda}}$ is defined using a particular ridge estimator associated with regularization term based on $\mathbf{\Lambda}$. In such a case, $\widehat{\theta}_{\text{luck}} = \widehat{\theta}_{\mathbf{\Lambda}}$ is a ridge estimator (see equations (8) and (15)).

2.5 Optimal redundancy as a complexity measure

One metric to measure the effectiveness of any code $\mathbb{Q}$ when the observations follow a generative model $\mathbf{y} \sim \mathbb{P}_\star$, is the *redundancy* or the expected excess code-length for $\mathbb{Q}$ compared to the true (but unknown) distribution $\mathbb{P}_\star$, given by:

$$\frac{1}{n}\mathbb{E}_{\mathbf{y}\sim\mathbb{P}_\star}\left[\log\left(\frac{1}{\mathbb{Q}(\mathbf{y})}\right) - \log\left(\frac{1}{\mathbb{P}_\star(\mathbf{y})}\right)\right] = \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_\star}\left[\frac{1}{n}\log\left(\frac{\mathbb{P}_\star(\mathbf{y})}{\mathbb{Q}(\mathbf{y})}\right)\right] = \frac{1}{n}\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_\star \,\|\, \mathbb{Q}), \quad (6)$$

where we normalized by n, to obtain bits(/nats) per sample point, since in our notation the code $\mathbb{Q}$ is defined jointly over the n observations. Here $\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_\star \,\|\, \mathbb{Q})$ denotes the Kullback-Leibler divergence between the two distributions and is non-negative unless $\mathbb{Q} = \mathbb{P}_\star$. Or, in other words, the unknown $\mathbb{P}_\star$ is also the best possible encoding of the data. Thus, the complexity of the data with respect to a given set of codes $\mathcal{Q}$ can be defined via the best possible excess codelength also known as *optimal redundancy*:

$$\mathcal{R}_{\mathrm{opt}}(\mathbb{P}_\star, \mathcal{Q}) := \min_{\mathbb{Q}\in\mathcal{Q}} \mathcal{D}_{\mathrm{KL}}(\mathbb{P}_\star \,\|\, \mathbb{Q}). \quad (7)$$

Remark: For the complexity measure (7) to be effective, the set of codes $\mathcal{Q}$ should be rich enough; otherwise, the calculated complexity can provide trivial or loose estimates (e.g., as discussed above, in overparameterized setting when $\mathcal{Q}$ contains only the NML code and $\mathcal{Y}$ is unbounded). As noted earlier, the advantage of the MDL framework comes from the fact that the set $\mathcal{Q}$ does not have to be the usual parametric class used for computing maximum likelihood (and as noted above, the NML and LNML codes are strict global generalizations of the maximum likelihood principle). Furthermore, even without a generative model, one can consider a minimax notion of redundancy as a complexity measure Barron et al. (1998). In the sequel, however, we restrict our derivations to settings with a known generative model over the observed data and briefly discuss a minimax set-up in Appendix A.4.

3. Ridge-based minimum description length complexity

In this work, we define a new complexity measure, MDL-COMP, using the optimality principle (7) and a family of LNML codes, that are induced by ridge estimators. In a nutshell, the luckiness function (4) is inspired by the penalty used in ridge regression. Our choice of ridge-based LNML codes is informed by multiple reasons: First, we are interested in an operational definition of complexity, and hence we consider encodings that are associated with a computationally feasible set of estimators. Second, for the complexity to be informative, we prefer the set of codes to be rich enough, and thus the estimators that we consider should achieve good predictive performance. Ridge estimators are known to achieve the minimax performance with linear and kernel methods (Raskutti et al., 2011; Zhang et al., 2015; Dicker, 2016), and often provide competitive predictive performance in applied machine-learning tasks (Bernau et al., 2014).[2]

[2] We do not simply use the ordinary-least-squares-estimator based code—which in the NML framework would reduce the problem to a single NML code, and the Shtarkov complexity—for the following reasons: (i) it has poor performance in the over-parameterized regime with linear models, and for any setting of kernel regression (for infinite-dimensional kernels), and (ii) as shown earlier, the Shtarkov complexity would be infinite when $\mathcal{Y}$ is unbounded, in the overparameterized settings considered in the sequel.

We now define the details of these codes in Sec. 3.1 (besides providing a brief recap of ridge regression), and then formally define the MDL-COMP in Sec. 3.2.

3.1 Ridge-based LNML codes

As in the MDL literature, we design codes over response vectors $\mathbf{y} \in \mathbb{R}^n$ conditional on a fixed matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ of covariates. We briefly recap ridge regression estimators for linear models and kernel methods in Sec. 3.1.1, followed by the definitions of the corresponding LNML codes in Sec. 3.1.2 that underlie our definition of MDL-COMP in Sec. 3.2.

3.1.1 Background on ridge regression

For linear models, the generalized ℓ_2-regularized least square estimators, with the penalty $\theta^\top \mathbf{\Lambda} \theta$ for a positive definite matrix $\mathbf{\Lambda}$, is given by

$$\widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}) := \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \left(\frac{1}{2} \|\mathbf{y} - \mathbf{X}\theta\|_2^2 + \frac{1}{2} \theta^\top \mathbf{\Lambda} \theta \right) = (\mathbf{X}^\top \mathbf{X} + \mathbf{\Lambda})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{8}$$

A common choice for the regularization matrix is $\mathbf{\Lambda} = \lambda \mathbf{I}$. We also define some notation to be useful later on:

$$\mathbf{X}^\top \mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{U}^\top \quad \text{where} \quad \mathbf{D} = \operatorname{diag}(\rho_1, \ldots, \rho_d), \tag{9}$$

where for $d > n$, we use the convention that $\rho_i = 0$ if $i > n$. Here the matrix $\mathbf{U} \in \mathbb{R}^{d \times d}$ denotes the (complete set of) eigenvectors of the matrix $\mathbf{X}^\top \mathbf{X}$.

Next, we briefly describe kernel ridge regression. Consider a reproducing kernel $\mathcal{K}$ and the corresponding reproducing kernel Hilbert space (RKHS), i.e., for all $x \in \mathbb{R}^d$, $\mathcal{K}(x, \cdot) \in \mathbb{H}$ and for any $f \in \mathbb{H}$, we have $\langle f, \mathcal{K}(x, \cdot) \rangle_{\mathbb{H}} = f(x)$. In kernel ridge regression, given the data $\{(x_i, y_i)\}_{i=1}^n$, we need find an estimate $\widehat{f} \in \mathbb{H}$ such that $\widehat{f}(x_i) \approx y_i$. The corresponding estimate is given by

$$\widehat{f} := \operatorname*{argmin}_{f \in \mathbb{H}} \left(\frac{1}{2} \|\mathbf{y} - \mathbf{f}_1^n\|_2^2 + \frac{\lambda}{2} \|f\|_{\mathbb{H}}^2 \right) \quad \text{where} \quad \mathbf{f}_1^n := (f(x_1), \ldots, f(x_n))^\top, \tag{10}$$

and $\lambda > 0$ denotes a regularization parameter. The representer theorem for reproducing kernels implies that to solve (10) it suffices to consider functions of the form $f(\cdot) = \sum_{i=1}^n \beta_i \mathcal{K}(x_i, \cdot)$ for $\beta \in \mathbb{R}^n$. Substituting this functional form back in equation (10) yields a regularized least-squares problem from θ which admits a closed-form:

$$\widehat{\beta}_\lambda(\mathbf{y}) := \operatorname*{argmin}_{\beta \in \mathbb{R}^n} \left(\frac{1}{2} \|\mathbf{y} - \mathbf{K}\beta\|_2^2 + \frac{\lambda}{2} \beta^\top \mathbf{K} \beta \right) = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}. \tag{11}$$

where $\mathbf{K}$ is the $n \times n$ kernel matrix with $\mathbf{K}_{ij} = \mathcal{K}(x_i, x_j)$. With this fit, one can recover the estimates for in sample observations as $\widehat{\mathbf{y}} = \mathbf{K}\widehat{\beta}_\lambda$, and for any new point x, the estimate is given by $\widehat{f}(x) = \sum_{i=1}^n [\widehat{\beta}_\lambda]_i \mathcal{K}(x_i, x)$.

3.1.2 Defining ridge-based LNML codes

We start with the linear model setting and then discuss the kernel setting. In accordance with MDL convention, we assume that the design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ is known, and only the information in the response vector $\mathbf{y} \in \mathbb{R}^n$ is to be encoded.

LNML codes for linear methods: We use the notation:

$$g(\mathbf{y};\mathbf{X},\mathbf{\Lambda},\theta)=\frac{1}{(2\pi\sigma^2)^{n/2}}e^{-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{X}\theta\|^2}\cdot e^{-\frac{1}{2\sigma^2}\theta^\top\mathbf{\Lambda}\theta},\quad \text{for}\quad \mathbf{y}\in\mathbb{R}^n,\theta\in\mathbb{R}^d. \tag{12}$$

Next, we define the first term in equation (12) as the code p_{data} for data fit, and the second term ($e^{-\frac{1}{2\sigma^2}\theta^\top\mathbf{\Lambda}\theta}$) as the luckiness factor, i.e.,

$$\begin{aligned} p_{\text{data}}(\mathbf{y};\mathbf{X},\theta)&:=\frac{1}{(2\pi\sigma^2)^{n/2}}e^{-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{X}\theta\|^2}\quad\text{and}\\ p_{\text{luck}}(\theta)=p_{\mathbf{\Lambda}}(\theta)&:=e^{-\frac{1}{2\sigma^2}\theta^\top\mathbf{\Lambda}\theta}\end{aligned} \tag{13}$$

and comparing with equation (4), we define the LNML code $\mathbb{Q}_{\mathbf{\Lambda}}$ as follows: $\mathbb{Q}_{\mathbf{\Lambda}}$ is the a distribution over $\mathbb{R}^n$, that admits the density q_{LNML} given by

$$q_{\text{LNML}}(\mathbf{y})=q_{\mathbf{\Lambda}}(\mathbf{y})=\frac{g(\mathbf{y};\mathbf{X},\mathbf{\Lambda},\widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}))}{\mathrm{C}_{\mathbf{\Lambda}}}\quad\text{where}\quad \mathrm{C}_{\mathbf{\Lambda}}:=\int_{\mathbb{R}^n}g(\mathbf{z};\mathbf{X},\mathbf{\Lambda},\widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{z}))d\mathbf{z}. \tag{14}$$

It is a valid density for an LNML code, denoted by $\mathbb{Q}_{\mathbf{\Lambda}}$, since

$$\operatorname*{argmax}_{\theta}\left(p_{\text{data}}(\mathbf{y};\mathbf{X},\theta)\cdot p_{\mathbf{\Lambda}}(\theta)\right)\overset{(8)}{=}\widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}). \tag{15}$$

In other words, $\mathbb{Q}_{\mathbf{\Lambda}}$ is the LNML code induced by the ridge estimator $\widehat{\theta}_{\mathbf{\Lambda}}$.[3] We note that the RHS of equation (12) denotes (up to proportionality) the posterior density on θ for Gaussian likelihood on data with a Gaussian prior; however, the distribution $\mathbb{Q}_{\mathbf{\Lambda}}$ is not central to the Bayesian settings.

Our definition of complexity, as alluded to earlier, makes use of a family of LNML codes. For the linear setting, we consider the family:

$$\mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}:=\{\mathbb{Q}_{\mathbf{\Lambda}}:\mathbf{\Lambda}\in\mathcal{M}\},\ \text{where}\ \mathcal{M}:=\left\{\mathbf{U}\,\mathrm{diag}(\lambda_1,\ldots,\lambda_d)\mathbf{U}^\top|\lambda_j\geq 0,j=1,\ldots d\right\}, \tag{16}$$

where $\mathbf{U}$ denotes the eigenvectors of the matrix $\mathbf{X}^\top\mathbf{X}$ (9). We note that the discussion to follow depends on the choice of luckiness functions as well as the consequent choice of $\mathcal{M}$. While our choice of luckiness function is akin to the choice of conjugate prior in Bayesian settings, namely for analytical tractability, the consequent choice of $\mathcal{M}$ makes a particular tradeoff between the codelength needed to encode data and the regularization parameters. See the discussion below, Sec. 3.3.3, and Remark 3.

Note that the family $\mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}$ is not parametric in the classical sense (as it is not indexed by the canonical parameter θ). However, since the LNML codes are induced by the ridge estimators, selecting a particular $\mathbb{Q}_{\mathbf{\Lambda}}$ code corresponds to choosing a particular $\mathbf{\Lambda}$ based-ridge estimator for all $\mathbf{y}$. Moreover, this family can be seen as a family of LNML codes induced due to different choices of the luckiness functions in equation (4).

A key difference from the NML setting is that the LNML codes are indexed by hyper-parameter $\mathbf{\Lambda}$, and thus it remains to add the codelength corresponding to the index. To

[3] In a similar fashion, one can see that the OLS estimator would correspond to the NML code (1) for the codes given by (3).

this end, we use a simple quantization-based encoding for the hyper-parameters $\{\lambda_j\}$, in the spirit of the two-stage encoding for the hyper-parameters as in prior works (Barron et al., 1998; Hansen and Yu, 2001; Grünwald, 2007). In particular, using $\mathcal{L}$ to denote the codelength corresponding the regularization parameters, we use

$$\mathcal{L}(\mathbf{\Lambda}) := \frac{1}{2} \sum_{\lambda_i > 0} \log \lambda_i, \tag{17}$$

where the factor $\frac{1}{2}$ is chosen for simplifying our analytical expressions in Theorem 1 and can replaced by 1 without changing the qualitative conclusions (see Appendix A.5).

Note that the expression (17) does not include any codelength to share the indices of $\{i : \lambda_i > 0\}$ for the following reasons. In the proof of Theorem 1, we show that $\lambda_i > 0$ if and only if $\rho_i > 0$ (for our analytical derivations for linear model settings). These indices are known both to sender and receiver since $\mathbf{X}$ is assumed known to both parties so the knowledge of what indices correspond to positive eigenvalues of $\mathbf{X}^\top \mathbf{X}$ is also apriori shared. Moreover in practice, we often only use one regularization parameter $\lambda_i = \lambda$ so that no additional coding of any index is needed. If one uses multiple λ_i's in practice as in equation (16), we can once again argue that for any practical choice, $\lambda_i = 0$ if an only if $\rho_i = 0$, so that no further addition of codelengths for indices is needed. We provide a further discussion of our choice of codelength for $\mathbf{\Lambda}$, and the set $\mathcal{M}$ in Sec. 3.3.

LNML codes for kernel methods: Given the estimators (10) and (11), one can define the LNML codes given a set of points, the kernel and the corresponding kernel matrix as follows. We define the function $h(\cdot; \beta, \mathbf{K})$

$$h(\mathbf{y}; \beta, \mathbf{K}) := \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{K}\beta\|^2} \cdot e^{-\frac{\lambda}{2\sigma^2}\beta^\top \mathbf{K}\beta}, \tag{18}$$

Letting the first term in equation (18) denote the code p_{data} for data fit, and the second term $(e^{-\frac{\lambda}{2\sigma^2}\beta^\top \mathbf{K}\beta})$ as the luckiness factor p_{luck}, and arguing similar to equations (14) and (15), we conclude that the LNML code is given by

$$\widetilde{q}_\lambda(\mathbf{y}) = \frac{h(\mathbf{y}; \widehat{\beta}_\lambda(\mathbf{y}), \mathbf{K})}{\mathrm{C}_{\lambda,\mathcal{K}}} \quad \text{where} \quad \mathrm{C}_{\lambda,\mathcal{K}} := \int_{\mathbb{R}^n} h(\mathbf{z}; \widehat{\beta}_\lambda(\mathbf{z}), \mathbf{K}) d\mathbf{z}, \tag{19}$$

and let $\widetilde{\mathbb{Q}}_\lambda$ denote the distribution corresponding to $\widetilde{q}_\lambda$. Finally, our definition of complexity for kernel methods makes use of the following family of LNML codes:

$$\mathcal{Q}^{\mathbf{K}}_{\text{Ridge}} = \left\{ \widetilde{\mathbb{Q}}_\lambda : \lambda \geq 0 \right\}. \tag{20}$$

In contrast to the linear setting, here we consider the family of codes indexed by a single parameter, as using multiple λ's would break the equivalence between equations (10) and (11), unless we alter the Hilbert norm that is being penalized in equation (10).

3.2 Defining MDL-COMP via ridge-LNML codes

We are now ready to define the MDL-based complexity. We define the complexity as optimal redundancy over the LNML codes with certain generative assumptions for the data.

MDL-COMP for linear models: We consider the generative model

$$y_i = x_i^\top \theta_\star + \xi_i, \quad \text{for} \quad i = 1, 2, \ldots, n, \quad \text{or} \quad \mathbf{y} = \mathbf{X}\theta_\star + \xi, \tag{21}$$

where we assume that $\xi \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$, so that $\mathbb{P}_\star = \mathbb{P}_{\theta_\star} = \mathcal{N}(\mathbf{X}\theta_\star, \sigma^2 \mathbf{I}_n)$.[4] Given this notation, the MDL–COMP for this setting is defined as the sum of the optimal redundancy over the codes $\mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}$ (16), and the codelength needed to encode the optimal hyperparameters:

$$\text{MDL–COMP}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}) := \frac{1}{n}(\mathcal{R}_{\text{opt}}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}) + \mathcal{L}(\mathbf{\Lambda}_{\text{opt}})), \quad \text{where} \tag{22a}$$

$$\mathcal{R}_{\text{opt}}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}) := \min_{\mathbb{Q} \in \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}} \mathcal{D}_{\text{KL}}(\mathbb{P}_{\theta_\star} \,\|\, \mathbb{Q}) = \min_{\mathbf{\Lambda} \in \mathcal{M}} \mathcal{D}_{\text{KL}}(\mathbb{P}_{\theta_\star} \,\|\, \mathbb{Q}_{\mathbf{\Lambda}}), \tag{22b}$$

and $\mathbf{\Lambda}_{\text{opt}}$ denotes the arg min in equation (22b).

Remark 1 *We note that for the definitions above, in principle, $\mathbb{P}_{\theta_\star}$ can be replaced by an arbitrary (and not necessarily even linear) generative model, and MDL-COMP would still be a valid complexity measure—as it measures the best possible excess number of bits over the class $\mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}$ for encoding data generated by $\mathbb{P}_{\theta_\star}$. However, for establishing analytical results in the following section, we restrict ourselves to a linear generative model $\mathbb{P}_{\theta_\star}$ while analyzing MDL-COMP with linear fitted models.*

MDL-COMP for kernel methods: We assume throughout this paper that the reproducing kernel $\mathcal{K}$ is a Mercer kernel (Mercer, 1909), which admits the eigen-expansion:

$$\mathcal{K}(x, y) = \sum_{k=1}^{\infty} \mu_k \phi_k(x) \phi_k(y), \quad \text{for all} \quad x, y \in \mathbb{R}^d, \tag{23}$$

where $\mu_1 \geq \mu_2 \cdots \geq 0$ denotes the sequence of (non-negative) eigenvalues of the kernel and $\{\phi_k\}_{k=1}^{\infty}$ denotes the associated eigenfunction taken to be orthonormal in $\mathbb{L}^2(\nu)$ for a suitably chosen distribution ν.[5] Let $\mathbb{H}$ denote the reproducing kernel Hilbert space of the kernel $\mathcal{K}$. We consider the generative model

$$y_i = f^\star(x_i) + \xi_i, \quad \text{for} \quad i = 1, 2, \ldots, n, \quad \text{or} \quad \mathbf{y} = (\mathbf{f}^\star)_1^n + \xi. \tag{24}$$

where we assume that x_i's are drawn i.i.d. from the distribution ν, and $f^\star \in \mathbb{H}$, and use the notation $(\mathbf{f}^\star)_1^n = (f^\star(x_1), \ldots, f^\star(x_n))^\top$. We also assume that $\xi \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$, so that $\mathbb{P}_\star = \mathbb{P}_{f^\star} = \mathcal{N}((\mathbf{f}^\star)_1^n, \sigma^2 \mathbf{I}_n)$.[6] With this notation, we define MDL-COMP for the kernel setting as the optimal redundancy over the codes $\mathcal{Q}^{\mathbf{K}}_{\text{Ridge}}$ in the family (20):

$$\text{MDL–COMP}(\mathbb{P}_{f^\star}, \mathcal{Q}^{\mathbf{K}}_{\text{Ridge}}) := \frac{1}{n} \min_{\widetilde{\mathbb{Q}} \in \mathcal{Q}^{\mathbf{K}}_{\text{Ridge}}} \mathcal{D}_{\text{KL}}(\mathbb{P}_{\theta_\star} \,\|\, \widetilde{\mathbb{Q}}) := \frac{1}{n} \min_{\lambda \geq 0} \mathcal{D}_{\text{KL}}(\mathbb{P}_{f^\star} \,\|\, \widetilde{\mathbb{Q}}_\lambda). \tag{25}$$

[4] Although we later discuss a minimax setting while relaxing this assumption on the noise, see Theorem 5.

[5] In the model below, we assume ν is the marginal distribution of the covariates.

[6] We can relax this assumption (without altering the guarantees derived later) to the noise being zero mean, with variance σ^2 and being uncorrelated with $(\mathbf{f}^\star)_1^n$.

where $\mathbf{K} = (\mathcal{K}(x_i, x_j))_{j=1}^n$ denotes the kernel matrix at the observed covariates. Like in the linear setting, one can replace $\mathbb{P}_{f^\star}$ by an arbitrary generative model and MDP-COMP would continue to be a valid measure of complexity. But for the analytical derivations in the sequel we restrict our attention to generative models of the form (24) when dealing with kernel methods. Notably, unlike in equation (22a), we do not add a codelength for λ in equation (25). See

3.3 Further discussion on MDL–COMP

In this section, we provide additional discussion and put the different choices made in our definitions in the context of prior work.

3.3.1 Relation of LNML code (15) with prior work

By definition (15), we have

$$-\log(q_{\mathbf{\Lambda}}(\mathbf{y})) = -\log\left(p_{\text{data}}(\mathbf{y}; \mathbf{X}, \widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}))\right) - \log(p_{\mathbf{\Lambda}}(\widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}))) + \log \mathrm{C}_{\mathbf{\Lambda}}. \tag{26}$$

Notably, the expression on the RHS of equation (26) is identical to that in Grünwald and Roos (2019, Eq. (6)), which is derived as the quantity of interest that characterizes the goodness-of-fit with the complexity of the MDL estimator. Moreover, same expression arises for a more general framework in Grünwald and Mehta (2017, Eqn. (52), arxiv version), while considering regularized ERM estimators under a unified treatment of MDL and several standard complexities.[7] However in both these works, analytical expressions for linear (or kernel) models with squared loss especially for overparameterized models are not considered. Here to make progress, we use the expected value of the quantity $-\log(q_{\mathbf{\Lambda}}(\mathbf{y}))$ under the true distribution of the data to compute the redundancy (6), and then use the minimum possible redundancy over $\mathbf{\Lambda}$ to measure our complexity measure MDL–COMP. Overall, our choice for the LNML code (26) is consistent with several prior works, and here we further enhance the understanding of such a choice with several analytical and experimental investigations.

3.3.2 The need for a true generative model

We highlight that equation (26) relies only on a *posited linear model* for the observed data. It is only to define the redundancy expression in equation (6), we assume a *generative model* associated with a true parameter $\theta_\star$[8]. The latter assumption is necessary for analytical derivations, and in the sequel we assume a linear generative model. However, for a given dataset our framework does not really require a generative model, as indeed equation (26) does not rely on a true linear model. Thus in practice, in accordance with the MDL principle, we directly minimize the codelength (26) over the choices of $\mathbf{\Lambda}$, and call it the practical version of MDL (Prac-MDL-COMP) that is also used in our experiments to tune the ridge regularization hyper-parameter (see Sec. 5 and equation (34)). As noted above such a recommendation is also consistent with the recommendation made in Grünwald and Roos

[7] We were made aware of these works by the reviewers.

[8] The subsequent calculations for the linear setting assume a generative linear model but equation (6) assumes just some generative model.

(2019, Eq. (6)) for selecting the best MDL-estimator when tuning over finitely many choices of $\mathbf{\Lambda}$ with a uniform prior over $\mathbf{\Lambda}$.

3.3.3 Regularization set $\mathcal{M}$: The tradeoff between expressivity and codelength

Our definition (16) makes use of the eigenvectors of the matrix $\mathbf{X}^\top\mathbf{X}$ to define the set $\mathcal{M}$ of all possible $\mathbf{\Lambda}$ that we consider for the LNML codes. In simple words, we assume that the regularization matrix $\mathbf{\Lambda}$ and the covariance matrix $\mathbf{X}^\top\mathbf{X}$ are simultaneously orthogonally diagonalizable. Such a choice tries to address two concerns: First, having a rich set of codes for analytical derivations is essential to provide us a better understanding of how much compression in the data is possible, so that having richer set than $\{\lambda\mathbf{I}; \lambda \geq 0\}$ (the common choice in practice) is desirable. On the other hand, if we make the set $\mathcal{M}$ too large, e.g., if it is the set of all positive semi-definite matrices, then we would defeat the purpose of measuring the complexity of data, as the bits needed to encode an arbitrary PSD matrix for linear model scale as $\mathcal{O}\left(\min\left\{d^2, n^2\right\}\right)$, which would just overwhelm the bits needed to encode the data itself. In the prior works with MDL, while deriving analytical expressions, much simpler choices like $\mathbf{\Lambda} = c\mathbf{X}^\top\mathbf{X}$ have been made (Hansen and Yu, 2001), and the bits needed to transmit the scalar c have been treated as fixed (as we do in equation (25) and Sec. 4.2 for kernel methods when we only have one hyper-parameter; also see Remark 2).

For our choice of $\mathcal{M}$, since matrix $\mathbf{U}$ and the indices $\mathcal{I} := \{j : \lambda_j > 0\}$ can be computed from $\mathbf{X}^\top\mathbf{X}$, we only need to count the bits needed to encode the hyper-parameters $\{\lambda_j, j \in \mathcal{I}\}$, for which we use equation (17). Such a choice allows more flexibility in the fitted model without blowing up the codelength for the hyperparameter too quickly. (We note that other careful choices of $\mathcal{M}$ are also possible; also see Remark 3.)

3.3.4 Our LNML encoding vs one-part universal encoding

For analytical derivations, we allow $\mathbf{\Lambda}$ to belong to a continuous family in equation (16). In such a setting, as noted in Grünwald and Roos (2019, Sec. 2.3, Eqns. (17,18)), we can associate a code length for $\mathbf{\Lambda}$ as well to define a *meta universal* (LNML) code.[9] where in the density (4) would be replaced by

$$q(\mathbf{y}) = \frac{\max_{\theta,\mathbf{\Lambda}} p_{\text{data}}(\mathbf{y}; \mathbf{X}, \theta) \cdot p_{\mathbf{\Lambda}}(\theta) \cdot \pi(\mathbf{\Lambda})}{\int_{\mathbb{R}^d} \max_{\theta',\mathbf{\Lambda}'}(p_{\text{data}}(\mathbf{z}; \mathbf{X}, \theta') \cdot p_{\mathbf{\Lambda}'}(\theta') \cdot \pi(\mathbf{\Lambda}'))d\mathbf{z}} \tag{27}$$

for a suitable luckiness function π over the space of $\mathbf{\Lambda}$ being considered. Such a one-part code was introduced for unifying model selection and estimation in Grünwald and Roos (2019, Sec. 2.3, Eqns. (17,18)) as it jointly tries to identify the best possible θ (estimation) as well as the regularization parameter $\mathbf{\Lambda}$ (model selection). With such a choice, the codelength would be indexed by the choice of the luckiness function π on the matrices $\mathbf{\Lambda}$, that we have to choose. (Notably, for the LNML code (27) to be well-defined, the π should be such that the denominator in display (27) is finite; ootherwise we still have an infinity problem like for the NML in equation (1).)

[9] We were made aware of the recent one-part meta-universal distribution framework by the reviewers during the review process.

Instead of searching for a suitable prefix code on $\mathbf{\Lambda}$, our treatment for MDL in equation (22a) instead involves adding a codelength after optimizing redundancy in equation (22a) using a simple quantization scheme (17). While a convenient choice for analytical calculations, this choice is also well-motivated by practical scenarios, when one typically deals with finitely many $\mathbf{\Lambda}$, in which case a uniform prior over them, i.e., $\pi(\mathbf{\Lambda}) \propto 1$ is a reasonable choice (Grünwald and Roos (2019)). Nevertheless in the continuous case a uniform prior is not well-defined and thus our choice warrants different interpretations: (A) using an improper luckiness function $\pi(\mathbf{\Lambda}) \equiv 1$ so that the expression from equation (27) degenerates to equation (4) and posthoc compensation with the codelength (17), or (B) as a direct analog of the treatment of encoding hyperparameters in crude two-stage MDL (Barron et al., 1998; Hansen and Yu, 2001; Grünwald, 2007). On the other hand, our theoretical results also provide two indirect justifications for our choice: The $\mathbf{\Lambda}_{\text{opt}}$ achieving the minimum in equation (22b) is indeed an *optimal* choice of regularization matrix $\mathbf{\Lambda}$ on two ends: (a) it minimizes the in-sample mean squared error (Theorem 2), and (b) it achieves the minimax codelength over a class of noise distributions with bounded variance (Theorem 5).

3.3.5 Codelength for discretized $\mathbf{\Lambda}$

Our codelength for $\mathbf{\Lambda}$ in equation (17) is in fact an approximation itself. In principle, we want to encode λ_i as an integer $\lceil \lambda_i/\Delta \rceil$ for some small enough resolution Δ; and $\Delta = \frac{1}{\sqrt{n}}$ is often the default choice. To encode integers, Rissanen (1983) shows that the best possible universal codelength takes the form $\log^\star \lceil \lambda_i/\Delta \rceil + C$, where $C \approx 1.52$ is a universal constant and for any integer $k \in \mathbb{N}$, we have $\log^\star(k) := \log_2 k + \log_2 \log_2 k + \cdots$ Lee (2001, Eqn. (4)). For the linear models, using this exact universal codelength would yield to an additive term of order $\frac{d}{2n} \log n$ in MDL–COMP expressions in Theorem 1 for $d < n$, and $\log n$ for $d > n$. As the typical scaling of MDL is indeed of order d/n for $d < n$, and $\log d$ for $d > n$ (see Fig. 1 and Appendix A.2), such an adjustment does not alter the scaling suggested by the current MDL–COMP expressions, and hence we continue to use the approximate $\log \lambda_i$ codelength in our subsequent discussion.

Remark 2 *For the kernel setting, we do not add another λ dependent codelength in equation (25), as there is only one regularization parameter in kernel regression; such a treatment is akin to the assuming fixed number of bits for transmitting the scalar c in prior works with MDL when using simpler choices like $\mathbf{\Lambda} = c\mathbf{X}^\top\mathbf{X}$ (see (Hansen and Yu, 2001). Moreover, adding a term $\log(\lambda_{opt})/n$ on the RHS of equation (25) does not alter the qualitative conclusions in Theorem 3 or quantitative conclusions in Corollary 1.*

Remark 3 *Our choices, namely (a) the luckiness function, (b) the class of hyper-parameters in the luckiness function, and (c) how we encode these hyper-parameters directly impact our complexity calculations to follow. For example, a different choice of $p_{\mathbf{\Lambda}}$, or $\mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}$, or a different codelength (instead of equation (17)) would lead to a different complexity measure; and the true complexity should be defined as the infimum complexity across all possible choices. Thus, our complexity measure (*MDL–COMP*) should be viewed as an upper bound on the true MDL complexity, which can be loose, however, as we demonstrate in the sequel, it is tighter than the naive parameter count when the linear model is overparameterized.*

4. Main results

We are now ready to state our main results. We start with an explicit characterization of MDL-COMP for linear models, and its consequences in Sec. 4.1. We then characterize it for kernel methods and unpack the consequences in Sec. 4.2.

4.1 Characterizing MDL-COMP for linear models

Our first result provides an explicit expression for MDL-COMP (22a) for the linear models.

Theorem 1 *For the linear model (21), let* $\mathbf{U}$ *and* $\{\rho_i\}$ *denote the eigenvectors and eigenvalues of* $\mathbf{X}^\top\mathbf{X}$*, define the vector* $\mathbf{w} := \mathbf{U}^\top\theta_\star$*, and recall that* σ^2 *denotes the noise variance. Then the MDL complexity (22a) and the optimal redundancy (22b) are given by*

$$\text{MDL–COMP}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}) = \frac{1}{2n}\sum_{i=1}^{\min\{n,d\}} \log\left(\rho_i + \frac{\sigma^2}{w_i^2}\right), \quad \textit{and} \tag{28a}$$

$$\mathcal{R}_{opt}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}) = \frac{1}{2n}\sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i w_i^2}{\sigma^2}\right). \tag{28b}$$

See Appendix C.1 for the proof. We recall that the NML complexity for over-parameterized settings is typically infinite, or just a function of the volume of the space when one truncates the space of observation for theoretical analysis (Sec. 2.3). Even in the latter case, for overparameterized setting, the complexity does not depend on the design matrix. Our notion of MDL-COMP on the other hand, as seen by Theorem 1, is not merely a parameter count or a simple function of d and n. Rather, it depends on the interaction between the eigenvalues of the covariance matrix $\mathbf{X}^\top\mathbf{X}$, and the rotated true parameter scaled by noise variance $\mathbf{w}/\sigma = \mathbf{U}^\top\theta_\star/\sigma$. The expression (28a) is oracle in nature since it depends on an unknown quantity, namely the true parameter $\theta_\star$ via the relation $\mathbf{w} = \mathbf{U}^\top\theta_\star$.[10]

In Sec. 5, we propose a data-driven and MDL–COMP inspired hyper-parameter selection criterion called Prac-MDL-COMP to tune the ridge hyper-parameter. Later in Sec. 5, we provide a data-driven approximation to MDL–COMP so that our proposed complexity can also be computed as a practical complexity measure without requiring knowledge of $\theta_\star$.

Next, we discuss some consequences for linear models: We illustrate the scaling of MDL–COMP in various settings in Sec. 4.1.1, and then prove in Sec. 4.1.2 that MDL–COMP informs the fixed design generalization error (see Theorem 2). Furthermore, in Appendix A.4, we establish a certain minimax optimality property of the code that defines MDL–COMP (see Theorem 5). We turn to MDL-COMP for kernel methods in Sec. 4.2.

4.1.1 Scaling of MDL–COMP for various covariate designs

We now numerically compute MDL–COMP in several synthetic settings. Below we plot results for three different settings on the covariate design $\mathbf{X}$, and two different settings for

[10] When model is under-specified in terms of the features, i.e., $\mathbf{X}$ includes a subset of features needed to correctly specify the model (21), $\mathbf{w}$ is defined by considering the restricted version of $\theta_\star$; and when it is over-specified, i.e., $\mathbf{X}$ denotes a superset of features, $\mathbf{w}$ is defined by appending zeros to $\theta_\star$ as necessary. Refer to footnote 17, and Appendix A.1 for further discussion.

the true parameter. In all cases, the rows of $\mathbf{X}$ are drawn from $\mathcal{N}(0, \Sigma)$. In Fig. 1, we consider two cases $\Sigma = \mathbf{I}_d$ (labeled as $\alpha = 0$), and $\Sigma = \mathrm{diag}(1, 2^{-\alpha}, \ldots, d^{-\alpha})$ for $\alpha = 0.5$. In Fig. 2, we choose a *spike design*, where $\Sigma = \mathrm{diag}(16, 16, \ldots, 16, 1, \ldots, 1)$ with the first s (spike dimension) diagonal entries taking value 16, and the rest taking value 1. In both figures, we evaluate two different settings of $d_\star$ for the true dimensionality of $\theta_\star$, and select the true parameter $\theta_\star$ by drawing i.i.d. entries from standard normal, and then normalizing it to have norm 1. Note that as we move from left to right on the x-axis in these figures, only the covariates used for fitting the model vary, and the generated data remains fixed (so that the model is under-specified for $d < d_\star$, and correctly (over) specified for $d \geq d_\star$). See Appendix A.1 for more details on the simulation set-up.

In both the figures, we note the non-linear scaling of MDL-COMP in the overparameterized regime ($d > n$). As we vary the dimension d of the covariates used for computing MDL–COMP, in Fig. 1, we observe a linear scaling of MDL–COMP with d for $d < n$, but a slow logarithmic or $\log d$ growth for $d > n$. On the other hand, the growth is clearly not linear even for $d < n$ for some of the spike design settings in Fig. 2. We provide further discussion on the set-up and scaling of MDL-COMP from these figures in Appendices A.1 to A.3 (also see Theorem 4).

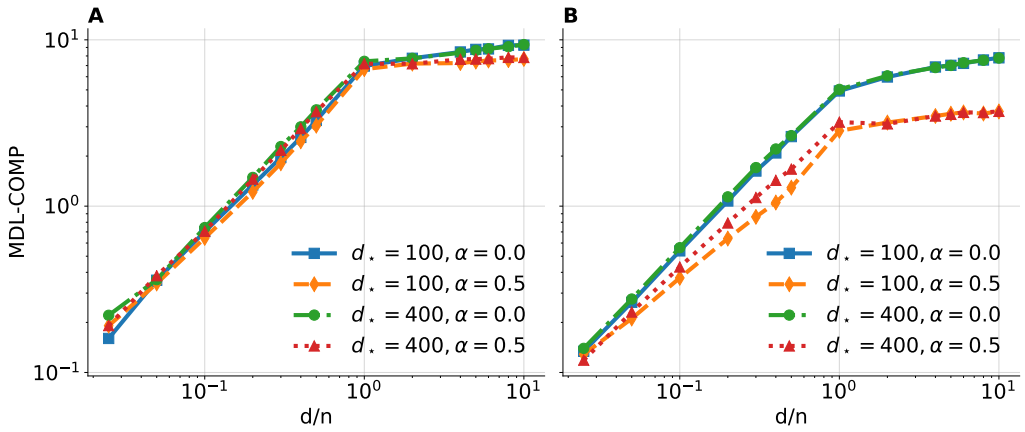

Figure 1. Scaling of MDL-COMP for Gaussian design. **(A)** $\sigma^2 = 1$, and **(B)** $\sigma^2 = 0.01$. We fix the generated data with $n = 200$ samples, and vary the dimensionality d of the covariates used for fitting the data. Here α denotes the decay of the eigenvalues in the covariance matrix for the covariates, and $d_\star$ denotes the true dimensionality of $\theta_\star$.

4.1.2 MDL–COMP informs fixed design mean-squared error

Next, we show that MDL–COMP (without the codelength for $\mathbf{\Lambda}$) directly bounds the optimal fixed design prediction error (MSE). For the training data points $\{(x_i, y_i)\}_{i=1}^n$ generated from a true model (21) with true parameter $\theta_\star$, the fixed design prediction MSE of an estimator $\widehat{\theta}$ is given by

$$\text{fixed design pred. MSE} := \frac{1}{n}\sum_{i=1}^{n}(x_i^\top \widehat{\theta} - x_i^\top \theta_\star)^2 = \frac{1}{n}\|\mathbf{X}\widehat{\theta} - \mathbf{X}\theta_\star\|^2. \tag{29}$$

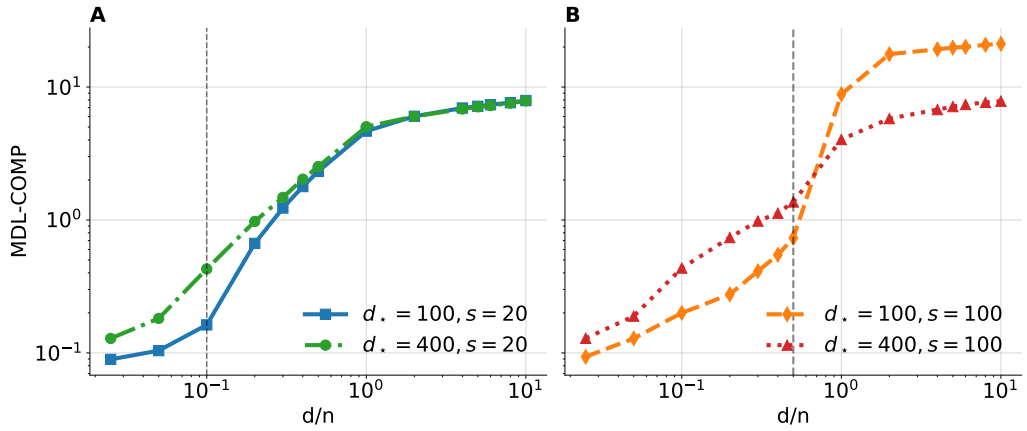

Figure 2. Scaling of MDL-COMP for spike design. We fix the generated data with $n = 200$ samples, and $\sigma^2 = 0.01$, and vary the dimensionality d of the covariates used for fitting the data. Here s denotes the spike dimension of the covariance matrix, and $d_\star$ denotes the true dimensionality of $\theta_\star$.

Note that this fixed design prediction error is very different from the training MSE ($\frac{1}{n}\|\mathbf{X}\widehat{\theta} - \mathbf{y}\|^2$). The in-sample MSE can be considered as a valid proxy for the out-of-sample MSE (which in turn is usually estimated by test MSE (33), when the out-of-sample points have the same covariates as that of the training data; and it has been often used in prior works to study the bias-variance trade-off for different estimators (Raskutti et al., 2014). Our next result shows that the optimal redundancy $\mathcal{R}_{\text{opt}}$ bounds the optimal in-sample MSE, and that $\mathbf{\Lambda}_{\text{opt}}$ that achieves $\mathcal{R}_{\text{opt}}$ and defined MDL–COMP also minimizes the in-sample MSE. The reader should recall the definition (22b) of $\mathcal{R}_{\text{opt}}$ and $\mathbf{\Lambda}_{\text{opt}}$.

Theorem 2 *For the ridge estimators* (8)*, we have*

$$\mathbb{E}_\xi\left[\frac{1}{n}\|\mathbf{X}\widehat{\theta}_{\mathbf{\Lambda}_{opt}} - \mathbf{X}\theta_\star\|^2\right] = \min_{\mathbf{\Lambda}\in\mathcal{M}} \mathbb{E}_\xi\left[\frac{1}{n}\|\mathbf{X}\widehat{\theta}_{\mathbf{\Lambda}} - \mathbf{X}\theta_\star\|^2\right] \tag{30a}$$

$$\leq \frac{\sigma^2}{n}\mathcal{R}_{opt}(\mathbb{P}_{\theta_\star}, \mathcal{Q}_{\text{Ridge}}), \tag{30b}$$

where $\mathbb{E}_\xi$ *denotes the expectation over the noise variables* $(\xi_1, \ldots, \xi_n)$ *from* (21)*.*

See Appendix C.2 for the proof of this claim.

Later, we show that in experiments, tuning the ridge model via a practical (data driven) variant of MDL–COMP can often minimize out-of-sample MSE, and provide predictive performance that is competitive with CV-tuned ridge estimator.

4.2 Characterizing MDL-COMP for kernel methods

We now turn to the non-linear settings, namely kernel methods. Our next result provides a bound on MDL–COMP for kernel methods.

Theorem 3 *For the kernel setting (24), the MDL complexity (25) is bounded as*

$$\text{MDL–COMP}(\mathbb{P}_{f^\star}, \mathcal{Q}^{\mathbf{K}}_{\text{Ridge}}) \le \inf_{\lambda} \left(\frac{\lambda}{2n} \frac{\|f^\star\|^2_{\mathbb{H}}}{\sigma^2} + \frac{1}{2n} \sum_{i=1}^{n} \log\left(\frac{\rho_i}{\lambda} + 1\right) \right) \tag{31}$$

where $\{\rho_i\}_{i=1}^n$ *denote the eigenvalues of the kernel matrix* $\mathbf{K} = (\mathcal{K}(x_i, x_j))_{i,j=1}^n$.

See Appendix C.3 for the proof. Note that unlike Theorem 1, here we establish an upper bound on MDL-COMP since closed-form expression does not exist. However, as we remark in the proof, we expect this bound to be tight for carefully constructed $f^\star$ for a wide range of behavior on $\{\rho_i\}$.

The bound (31) in Theorem 3 is generic, and can be applied to any kernel setting including the neural tangent kernels that have recently been investigated in the theoretical literature on deep neural networks. Like equation (28a), this expression also depends on the various problem parameters, including the signal-to-noise ratio $\frac{\|f^\star\|_{\mathbb{H}}}{\sigma}$, and the eigenvalues of the kernel matrix. When additional information on the decay of the eigenvalues $\{\rho_i\}$ is available, we can characterize a more refined bound on MDL–COMP as in the next corollary. For simplicity in exposition, we use $a_i \precsim b_i$ to denote that $a_i \le c b_i$ for all i with some universal constant c independent of i.

Corollary 1 *Suppose* $\mathcal{K}(x, x) = 1$, ρ_i *denote the eigenvalues of the kernel matrix* $\mathbf{K}$, *and* $\text{SNR} := \frac{\|f^\star\|_{\mathbb{H}}}{\sigma} > C$ *for some universal constant* C. *Then, we have*

$$\text{MDL–COMP}(\mathbb{P}_{f^\star}, \mathcal{Q}^{\mathbf{K}}_{\text{Ridge}}) \le \begin{cases} \dfrac{d \log^2(nd \cdot \text{SNR}^2)}{n} & \text{if } \rho_i \precsim n \exp(-i^{1/d}), \\ C_{d,\omega,\text{SNR}} \cdot \left(\dfrac{\log(n \cdot \text{SNR}^2)}{n}\right)^{\frac{2\omega}{2\omega+d}} & \text{if } \rho_i \precsim n i^{-2\omega/d}, \omega > d/2, \\ C_{d,a,\text{SNR}} \cdot \left(\dfrac{\log(n \cdot \text{SNR}^2)}{n}\right)^{\frac{d+a}{d+a+1}} & \text{if } \rho_i \precsim n i^{-d-a}, d+a > 1, \end{cases} \tag{32}$$

where the constants $C_{d,\omega,\text{SNR}}, C_{d,a,\text{SNR}}$ *are independent of* n, *and are defined in equation (73).*

See Appendix C.6 for the proof, where we also provide expressions for the constants appearing on the RHS above. We note that the n appearing in the scaling of the eigenvalues is not an arbitrary assumption, but an immediate consequence of the fact that the trace of the kernel matrix is equal to n since $\mathcal{K}(x, x) = 1$. We now contextualize the consequences of Corollary 1 compared to prior work on kernel methods, and some recent work on neural networks. Contrary to the linear model setting, here our discussion focuses primarily on the classical non-parameteric setting in low dimensions, in particular assuming $n \gg d$.

MDL–COMP **informs minimax in-sample MSE for kernel methods when** $n \gg d$ The eigenvalue decay rate of order $\exp(-i^{1/d})$ and, $i^{-2\omega/d}$ are known to be exhibited by Gaussian kernels and reproducing kernels (like Matérn kernels) for Sobolev spaces of smoothness ω in $\mathbb{R}^d$ respectively (see Santin and Schaback (2016, Thm. 15,16)). Up to logarithmic factors, the scaling of MDL–COMP in equation (32) with the sample size

$n \gg d$ for these settings matches with the minimax-optimal scaling of the in-sample risk $\frac{1}{n}\sum_{i=1}^{n}(f^\star(x_i)-\widehat{f}(x_i))^2$ (Stone, 1982; Raskutti et al., 2014; Wainwright, 2019). Consequently, in such cases, the scaling of MDL–COMP is directly informative of the minimax fixed design prediction error. This matching of rates between MDL–COMP and the minimax error provides an indirect justification for our choice of LNML codes (20) based on ridge estimators.

Applying MDL-COMP for neural tangent kernels We now briefly illustrate an example of how our theory can be used to approximately characterize the MDL-complexity for neural networks in certain settings. A suitable class of kernels, namely, neural tangent kernels (NTK) have been used in recent years to understand the theoretical properties of deep neural networks (Jacot et al., 2018; Hayou et al., 2019; Bietti and Mairal, 2019). In particular, it has been established that for certain scaling of parameters of a deep neural network, as the width of the network goes to infinity, the function represented by DNNs converges to that of a function in the RKHS of a suitable kernel. As expected, the nature of these kernels are governed primarily by the number of layers, the non-linear activation function of DNN, and the input data distribution. More recently, several works have established a spectral characterization of these kernels, e.g., the eigenvalue decays at the rate i^{-d-a} with $a = 0$ for deep neural tangent kernel (NTK) with ReLU activation functions, and $a = \frac{1}{2^L}$ for an L-layer deep NTK with step activation function (see Bietti and Bach (2021, Cor. 2,3)) when the covariates are drawn uniformly from the d-dimensional unit sphere. Combing these results with Corollary 1 readily yields the MDL complexity bounds for associated kernels. For instance, for NTK with ReLU activations, the bound (32) and the constant (73) from the proof show that the MDL-COMP for NTK with ReLU activations scales as $\mathrm{SNR}^{2/(1+d)} \cdot ((d\log n)/n)^{d/(d+1)}$, up to logarithmic factors. Thus we note that for a fixed but large sample size $n \gg d$, this complexity can decrease as the dimensionality d of the data increases, under the assumption that the change in SNR with dimension d does not alter the scaling with respect to n.

5. Experiments with data-driven MDL-COMP

This section proposes a practical version of MDL–COMP. Simulations and real-data experiments show that this data-driven MDL–COMP is useful for informing generalization. In the experiments to follow, this data-driven MDL–COMP as a hyperparameter tuning criteria. While Theorem 2 guarantees that the optimal regularization defining (the oracle) MDL-COMP also obtains the minimum possible in-sample MSE, in this section we numerically illustrate the usefulness of the Prac-MDL-COMP for achieving good test MSE which is computed on a fresh set of samples $(x_i', y_i')_{i=1}^{n_{\text{test}}}$ as follows:

$$\text{test-MSE} := \frac{1}{n_{\text{test}}}\sum_{i=1}^{n_{\text{test}}}(y_i' - \widehat{\theta}^\top x_i')^2. \tag{33}$$

We start by defining MDL–COMP inspired hyper-parameter tuning (model selection for regularization parameter) in Sec. 5.1 that we call Prac-MDL-COMP; see equation (34). Then, in Sec. 5.2 we demonstrate that solving (34) correlates well with minimizing test MSE (Fig. 3) for linear models. Next, in Sec. 5.3 we find that, for real datasets, MDL–COMP is competitive with cross-validation (CV) for tuning regularization hyperparameter, and actually

outperforming CV in the low-sample regime (Fig. 4). We note that CV is computationaly costlier to implement than since our method requires only training one model per choice (see end of Sec. 5.1). In Sec. 5.4, we evaluate our method on fMRI data and compare with two other baselines, Bayesian ARD and BIC, besides CV and find from Figs. 5 and B1 that Prac-MDL-COMP outperforms the ARD and BIC, and remains competitive with CV (note that BIC and Prac-MDL-COMP have same order of computational cost). We also provide comparison of CV and our method using neural tangent kernels in Fig. 7 of Sec. 5.5. Overall, we find that Prac-MDL-COMP provides a computationally efficient competitive alternative to CV for hyperpoarameter tuning for ridge regression (without losing on test error) across a range of real-world datasets.

In this section, the linear models (ridge) and kernel methods are fit using scikit-learn (Pedregosa et al., 2011) and optimization for hyper-parameter tuning (see (34)) is performed using SciPy (Virtanen et al., 2020). Code and documentation for easily reproducing the results are provided at github.com/csinva/mdl-complexity.

5.1 MDL–COMP inspired hyper-parameter tuning

As defined, the complexity MDL–COMP can not be computed in practice, since it assumes knowledge of true parameter. Moreover, ridge estimators are typically fit with only one regularization parameter, shared across all features. As an alternative that circumvents these issues, we propose the following data-driven *practical MDL-COMP*:

$$\text{Prac-MDL-COMP} = \min_{\lambda} \frac{1}{n} \left(\frac{\|\mathbf{X}\widehat{\theta}_\lambda - \mathbf{y}\|^2}{2\sigma^2} + \frac{\lambda \|\widehat{\theta}_\lambda\|^2}{2\sigma^2} + \frac{1}{2} \sum_{1=i}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i}{\lambda}\right) \right), \tag{34}$$

where $\widehat{\theta}_\lambda := \left(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^\top \mathbf{y}$ is the ridge estimator (8) for $\mathbf{\Lambda} = \lambda \mathbf{I}_d$, and we use ρ_i to denote the non-zero eigenvalues of the matrix $\mathbf{X}^\top \mathbf{X}$. This expression can be derived in two different ways: (i) The expression inside the minimizer is the n-sample "plug-in" estimate of the objective (22b) as shown in our proof in Theorem 1 in expression (49) (up to a constant offset 1/2), when we enforce the choice $\mathbf{\Lambda} = \lambda \mathbf{I}$. (ii) As discussed in Sec. 3.3, minimizing the LNML code length (26) to tune the hyper-parameter and find a good linear model fit is justified under a uniform prior over finitely many choices of hyper-parameter, a setting common in practice. Using the definitions (13), equation (26), and the expression for the normalization constant ($\mathrm{C}_{\mathbf{\Lambda}}$) in equation (50b) from the proof of Theorem 1 with $\mathbf{\Lambda} = \lambda \mathbf{I}$, we find that the objective (26) scaled by n can be simplified as follows:

$$-\frac{1}{n} \log(q_{\mathbf{\Lambda}}(\mathbf{y})) = -\frac{1}{n} \log\left(p_{\text{data}}(\mathbf{y}; \mathbf{X}, \widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}))\right) - \frac{1}{n} \log(p_{\mathbf{\Lambda}}(\widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y}))) + \frac{1}{n} \log \mathrm{C}_{\mathbf{\Lambda}} \tag{35}$$

$$= \frac{1}{n} \left(\frac{\|\mathbf{X}\widehat{\theta}_\lambda - \mathbf{y}\|^2}{2\sigma^2} + \frac{\lambda \|\widehat{\theta}_\lambda\|^2}{2\sigma^2} + \frac{1}{2} \sum_{1=i}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i}{\lambda}\right) \right), \tag{36}$$

which indeed is same as the objective in equation (34) used to define Prac-MDL-COMP.

Moreover, using equation (53) from the proof of Theorem 1, we can also obtain the plug-in estimate for optimal redundancy ($\mathcal{R}_{\text{opt}}$) as follows:

$$\widehat{\mathcal{R}}_{\text{opt}} := \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i}{\widehat{\lambda}_{\text{opt}}}\right), \tag{37}$$

where $\widehat{\lambda}_{\text{opt}}$ is the optimal hyperparameter for the objective (34). Since we only have one hyper-parameter in defining Prac-MDL-COMP, we can also treat the approximate $\widehat{\mathcal{R}}_{\text{opt}}$ as a proxy for MDL–COMP over the class $\{\mathbb{Q}_\lambda : \lambda \in (0, \infty)\}$.

For the same reasons as linear methods, a reasonable data-driven MDL criterion for tuning λ with kernel methods can be given by

$$\text{Prac-MDL-COMP}_{\mathbf{K}} = \min_{\lambda} \frac{1}{n}\left(\frac{\|\mathbf{K}\widehat{\theta}_\lambda - \mathbf{y}\|^2}{2\sigma^2} + \frac{\lambda \widehat{\theta}_\lambda^\top \mathbf{K} \widehat{\theta}_\lambda}{2\sigma^2} + \frac{1}{2}\sum_{i=1}^{n} \log\left(1 + \frac{\rho_i}{\lambda}\right)\right) \tag{38}$$

where for the kernel case, we use the notation $\widehat{\theta}_\lambda := (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}$ (10), and $\{\rho_i\}_{i=1}^n$ denote the eigenvalues of the kernel matrix $\mathbf{K}$.

Finally, while our theoretical results assume that that σ is known, in practice, often σ is unknown. In such a setting, we can estimate σ in the underparameterized as in least squares, namely, $\widehat{\sigma}^2 := \|\mathbf{X}\widehat{\theta}_{\text{OLS}} - \mathbf{y}\|^2/(n-d)$, where $\widehat{\theta}_{\text{OLS}} := (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$. Estimating the noise variance in the overparameterized settings is an active area of interest, and one possibility is to use a variance estimate from ridge regression with a suitable choice of hyperparameter (Liu et al., 2020, Eqn. (2)). Liu et al. (2020) establish a consistency and central limit theorem for this estimate under a linear generative model with high-dimensional asymptotics as $d/n \to \tau$ for $\tau \in [0, \infty)$.[11]

Computational benefits of Prac-MDL-COMP over cross-validation: We note that for a given hyperparameter λ, implementing the Prac-MDL-COMP criterion requires (1) solving a regularized least squares problem of size $n \times d$, and (2) computing the eigenvalues of the matrix $\mathbf{X}^\top\mathbf{X}$, both of which take $\mathcal{O}\left(dn^2\right)$ time when $d > n$, and $\mathcal{O}\left(nd^2\right)$ time when $n > d$ when using practical and stable numerical solvers. Thus the overall computational complexity is $\mathcal{O}\left(\min(n,d)^2 \max(n,d)\right)$. On the other hand, implementing k-fold cross-validation for a given λ requires us to solve k regularized least squares problem of size $\mathcal{O}(n) \times d$, and thereby the overall computational complexity is $\mathcal{O}\left(k \cdot \min(n,d)^2 \max(n,d)\right)$. In simple words, Prac-MDL-COMP is $\mathcal{O}(k)$ computationally more efficient than k-fold cross-validation.

5.2 Prac-MDL-COMP informs test MSE in Gaussian simulations

Fig. 3 shows that the model corresponding to the optimal λ achieving the minimum in equation (34), has a low test MSE (panels A-E), and comparable to the one obtained by leave-one-out cross validation (LOOCV, panel F). The setup follows the Gaussian model as in equation (21) with the noise variance σ^2 set to 1, with sample size $n = 100$ fixed. Here

[11] In all our simulations, we set $\sigma = 1$, same as ground truth. For the FMRI experiments, the results with $\sigma = 1$ were better than those obtained by estimating the variance in observations across 10 repetitions.

the covariates are drawn i.i.d. from $\mathcal{N}(0,1)$, and then fixed. The true parameter $\theta_\star$ is set to be in dimension 50 (extended to larger dimensions by appending zeros); its entries are first drawn i.i.d. from $\mathcal{N}(0,1)$ and then scaled so that $\|\theta_\star\| = 1$. We tune the parameter λ over 20 values equally spaced on a log-scale from 10^{-3} to 10^6.

We vary the number of covariates (d) used for fitting the model and report the results for $d/n \in \{1/10, 1/2, 1, 2, 10\}$ (noting that we have a misspecified model when fitting with $d < 50$ features). Across all panels (A-E), we observe that the minima of the test MSE and the objective (34) for defining Prac-MDL-COMP often occur close to each other (points towards the bottom left of these panels). Fig. 3F shows the generalization performance of the models selected by Prac-MDL-COMP in the same setting. Selection via Prac-MDL-COMP generalizes well, very close to the best ridge estimator selected by leave-one-out cross-validation. While the tuned ridge estimators (via CV, or MDL-COMP) exhibit the usual U-shaped curve for the test error in Fig. 3F, the OLS estimator exhibits a peak, a phenomenon termed as double-descent (that has been seriously investigated in recent works; see Sec. 6 for further discussion) .

Appendix B shows more results suggesting that Prac-MDL-COMP can select models well even under different forms of misspecification.

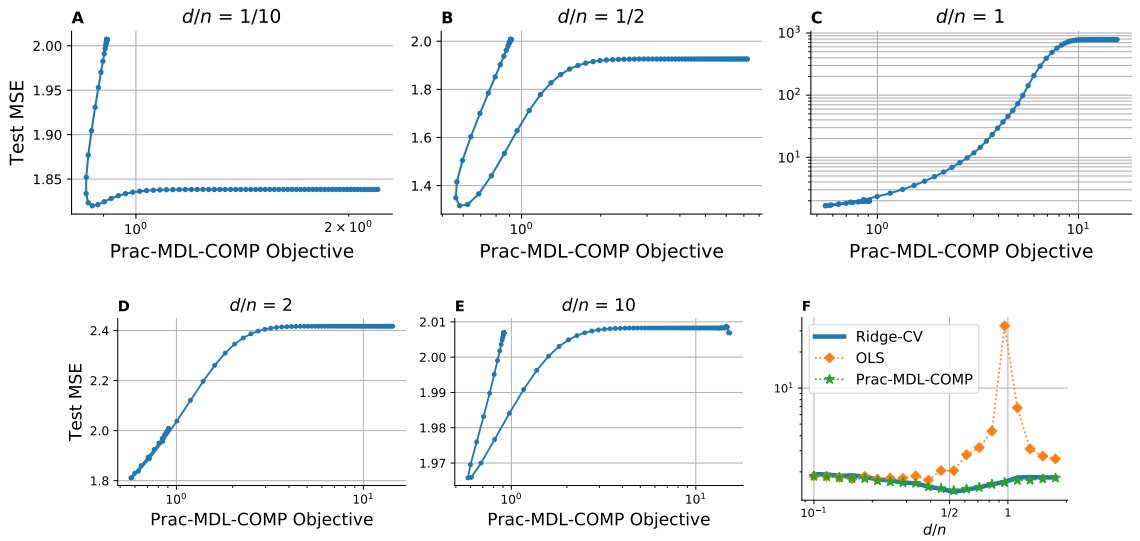

Figure 3. Minimizing the objective (34) which defines Prac-MDL-COMP selects models with low test error. **A-E.** Different panels show that this holds even as d/n is varied. **F**. Prac-MDL-COMP selects a model with Test MSE very close to Ridge cross-validation, avoiding the peak exhibited by OLS.

5.3 Experiments with PMLB datasets

In this section, we report results on the behavior Prac-MDL-COMP (34) when used to perform model selection on real datasets. Datasets are taken from PMLB (Olson et al., 2017; Vanschoren et al., 2013), a repository of diverse tabular datasets for benchmarking machine-learning algorithms. We omit datasets that are simply transformations of one another, or that have too few features; doing so yields a total of 19 datasets spanning a

variety of tasks, such as predicting breast cancer from image features, predicting automobile prices, and election results from previous elections (Simonoff, 2013). The mean number of data points for these datasets is 122,259 and the mean number of features is 19.1. For a given dataset, we fix d to be the number of features, and we vary n downwards from its maximum value (by subsampling the dataset) to construct instances with different values of the ratio d/n. The hyperparameter λ takes on 10 values equally spaced on a log scale between 10^{-3} and 10^{3}. The test set consists of 25% of the entire dataset.

Fig. 4A compares the performance of Prac-MDL-COMP to Ridge-CV. We find that shows that in the limited data regime, i.e. when d/n is large, Prac-MDL-COMP tends to outperform. As the number of training samples is increased (i.e., d/n decreases), the advantage provided by selection via Prac-MDL-COMP decreases. Further details, and experiments on omitted datasets are provided in Appendix B.2; in particular, see Figs. B2 and B4 and Table B1.

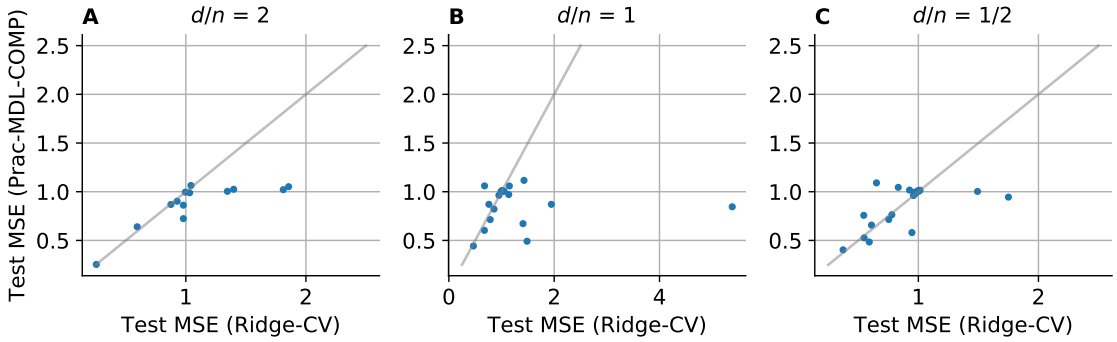

Figure 4. Comparing test-error when selecting models using Prac-MDL-COMP versus using cross-validation on real datasets. **A**. When data is most limited, Prac-MDL-COMP based hyperparameter tuning outperforms Ridge-CV. **B, C**. As the amount of training samples increases, Prac-MDL-COMP performs comparably to Ridge-CV. Each point averaged over 3 random bootstrap samples.

5.4 Experiments with fMRI data

We now focus on a challenging type of data that arises in neuroscience. The dataset consists of neural responses of human subjects, as recorded by functional magnetic-resonance imaging (fMRI), as they are shown natural movies (Nishimoto et al., 2011). The training data consists of 7,200 time points and the test data consists of 540 time points, where at each timepoint a subject is watching a video clip. The test data is averaged over 10 repetitions of showing the same clip to the same subject. Following the previous work, we extract video features using a Gabor transform, resulting in 1,280 features per timepoint. From these features, we predict the response for each voxel in the brain using ridge regression. To summarize, for this setting, we have $d = 1280, n_{\text{train}} = 7200$ and $n_{\text{test}} = 540$. We restrict our analysis to 50 voxels with no missing data in the V1, V2, and V4 regions of the brain, which are known to be easier to predict. Before fitting, the features and responses are each normalized to have mean zero and variance one. In all fMRI experiments, λ takes on 40 values equally spaced on a log scale between 10^{0} and 10^{6}.

Fig. 5 shows our prediction results when using Prac-MDL-COMP for model selection to predict the fMRI responses. We also compare our approach to Automatic Relevance Determination (ARD),[12] also known as Sparse Bayesian Learning, a popular Bayesian approach which places an adaptive prior over the regression parameters (MacKay, 1994) and BIC.[13] Fig. 5A and Fig. B1 respectively show that Prac-MDL-COMP consistently outperforms the Bayesian ARD baseline as well as BIC across voxels. Moreover, Prac-MDL-COMP is roughly on par with leave-one-out cross-validation (CV) for model selection in this data (Fig. 5). CV outperforms Prac-MDL-COMP for a majority of the voxels by a slight margin, but on the remaining voxels, Prac-MDL-COMP tends to outperform CV by a substantial margin.

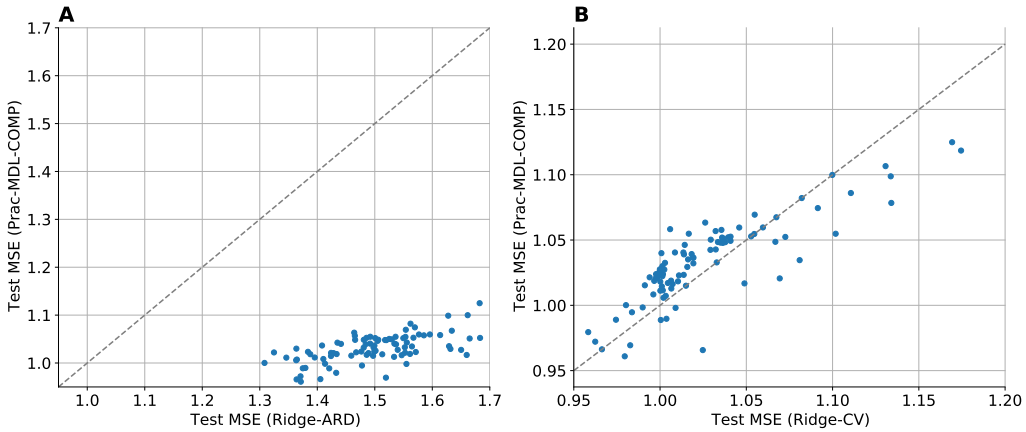

Figure 5. Prac-MDL-COMP succesfully selects models which predict fMRI responses well. **A**. Prac-MDL-COMP outperforms the Bayesian ARD baseline for every voxel (each point represents one voxel). Prac-MDL-COMP similarly outperforms a baseline using the BIC criterion (see Fig. B1). **B**. Prac-MDL-COMP is on par with cross-validation.

Fig. 6A shows the relationship between the Prac-MDL-COMP objective and the test error. Test error tends to increase as the Prac-MDL-COMP objective increases, showing that minimizing the objective continues to inform good model selection for minimizing test error even for this challenging real dataset. In Fig. 6B, we provide a scatter plot of the test MSE versus $\widehat{\mathcal{R}}_{\text{opt}}$ computed from data. We notice a linear relationship between the two quantities, and observe a correlation of 0.69. While, Theorem 2 guaranteed that the optimal redundancy $\mathcal{R}_{\text{opt}}$ bounds the in-sample MSE, Fig. 6B suggests that its data-driven approximation also provides useful information about the (ordering of) test MSE.

[12] Our choice to compare against ARD is also governed by the fact that ARD places a centered elliptic Gaussian distribution of the weights w; this means each coefficient w_i can be drawn from a Gaussian distribution centered on zero with a unique covariance matrix. This leads to sparser coefficients w, a natural setting for the fMRI prediction problem we study.

[13] In contrast to Prac-MDL-COMP's objective (34), the BIC objective for tuning λ is $\min_\lambda \frac{1}{n}\left(\frac{\|\mathbf{X}\widehat{\theta}_\lambda - \mathbf{y}\|^2}{2\sigma^2} + \frac{\log n}{2}\sum_{i=1}^{\min\{n,d\}} \frac{\rho_i}{\rho_i+\lambda}\right)$.

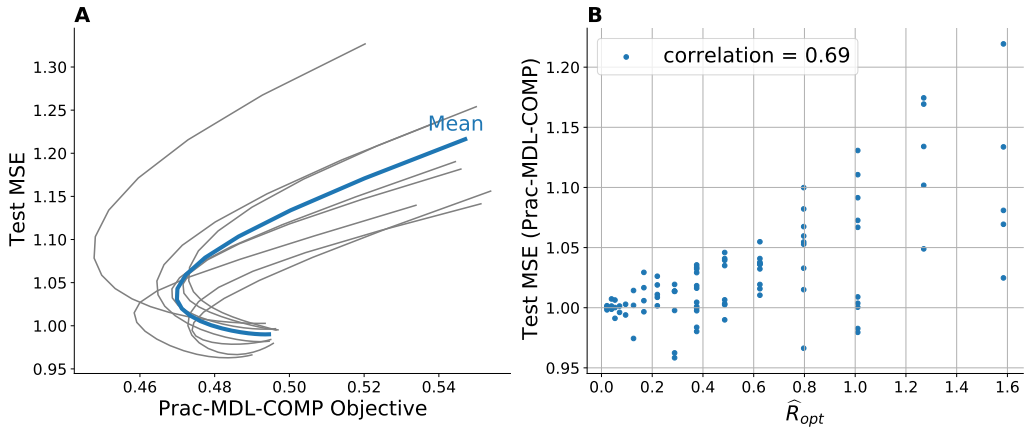

Figure 6. A. Test MSE often increases when Prac-MDL-COMP objective increases, showing that minimizing the objective informs good model selection for minimizing Test MSE. Each gray curve represents one randomly chosen voxel (only 10 out of 50 are shown for visualization). **B** Scatter plot (over the 50 different voxels) of test MSE versus the estimated $\widehat{\mathcal{R}}_{\text{opt}}$.

5.5 Experiments with neural tangent kernel on fMRI data

Fig. 7 shows the results when repeating the experiments in Fig. 5, but now using kernel ridge regression with the neural tangent kernel (Jacot et al., 2018) rather than linear ridge regression. It shows the prediction results when using Prac-MDL-COMP (applied to kernel ridge regression, see equation (38)) for model selection to predict the fMRI responses. For the neural tangent kernel computation, we use the neural-tangents library (Novak et al., 2020) with its default parameters (ReLU nonlinearity, two hidden linear layers with hidden size of 512). Fig. 7A shows that Prac-MDL-COMP is roughly on par with leave-one-out cross-validation for model selection in this data. Fig. 7B shows the relationship between the Prac-MDL-COMP objective and the test error, where we see that the curves are flatter when compared to Fig. 6A. Nonetheless, the test error often decreases as the Prac-MDL-COMP objective increases, suggesting that minimizing the objective is a good proxy for model selection for minimizing test error even for this setting.

6. Discussion

In this work, we revisited complexity measures in the context of overparameterized models. We argued that there is a lack of theoretical justification for using parameter count as a complexity measure in overparameterized settings. We defined an MDL-based complexity measure MDL-COMP for linear and kernel methods, using codes induced by ridge estimators that can also deal with overparameterized settings. Our analytical results show that MDL-COMP depends on dimension d, sample size n, the covariate/kernel matrix, the true parameter/function properties, and the noise in the data. It does not grow linearly in d for over-parameterized linear models, in fact, it often grows much more slowly as $\log d$ for $d > n$. We prove that MDL-COMP informs the fixed design generalization error and provides good empirical results for the random design generalization error. Numerical experiments show that a practical hyperparameter tuning scheme, inspired by MDL-COMP, provides

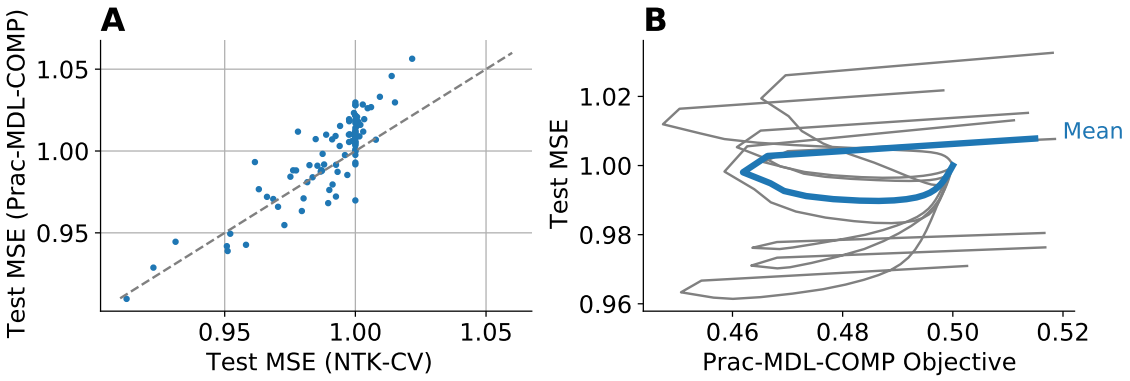

Figure 7. Prac-MDL-COMP succesfully selects models which predict fMRI responses well when using NTK-kernel. **A**. Prac-MDL-COMP is on par with cross-validation for every voxel (each point represents one voxel). **B**. Test MSE often increases when Prac-MDL-COMP objective increases, showing that minimizing the objective informs good model selection for minimizing Test MSE. Each gray curve represents one randomly chosen voxel (only 8 are shown for visualization).

generalization performance for various types of ridge estimators, often (although not always) outperforming cross-validation (CV) when the number of observations is limited. Moreover, MDL-COMP based tuning offers computational savings over K-fold cross validation as it tunes the parameter using only a single-fold computation, thereby serving as a competitive alternative to CV in limited data regimes.

6.1 Consequences for bias-variance tradeoff in overparameterized models

Another consequence of our results is a better understanding of the recent mysteries around the bias-variance tradeoff principle with overparameterized models which we now elaborate. While the classical bias-variance tradeoff is known to exhibit a U-shaped curve in well-posed regimes, several recent works have exhibited a *double-descent* curve that looks like the peaky curve of the OLS estimator from Fig. 3F (also see Fig. F1(a)) in ill-posed regimes. (Notably we do not observe the double descent with good regularized estimators.) Such a double-descent behavior for test error—with parameter count like measures on the x-axis as a proxy for complexity—has been observed and investigated in a series of recent works for linear regression (Hastie et al., 2019; Belkin et al., 2019b; Muthukumar et al., 2020) as well as DNNs for classifications (Advani and Saxe, 2017; Nakkiran et al., 2019) among other models (Belkin et al., 2019a).[14]

Investigating two possible causes of double descent: Our investigation into the complexity measures was partially motivated to better understand the double descent phenomenon. First, we note that the classical bias-variance tradeoff generally applies to

[14]It is worth noting that this double-descent phenomenon was documented as instability in early work on linear discriminant analysis (Skurichina and Duin, 1998, 2001, 2002). See Loog et al. (2020) for further discussion on the history of double descent. These non-classical test error curves have prompted several researchers to question the validity of the bias-variance tradeoff principle, especially in overparameterized regimes (Belkin et al., 2019a,b). We highlight that double descent can also occur in non-overparameterized regimes when the covariate matrix is ill-conditioned (see Fig. F1(c), and the discussion in this section).

a fixed training/test dataset in well-posed regimes, with a class of good estimators that are ordered by a suitable notion of complexity. Thus, a non-classical test error curve, in principle, can arise for two reasons: (i) The choice of complexity measure is not suitable or well-justified, e.g., the parameter count in overparameterized regimes. (ii) The choice of estimators is not suitable due to ill-posedness of the data/model, e.g., OLS estimators with ill-conditioned covariate matrix, or in overparameterized regimes. The concurrent work on double descent uses parameter count as the complexity measure while plotting the test error of a range of ill-posed and well-posed estimators or models[15] (the shape of the OLS curve in Fig. 3F is representative of the double descent figures in the recent related work), and thus apriori it is not clear if the double descent phenomenon is a consequence of (i) or (ii).

As a check for (i) (complexity), we can replace parameter count with MDL-COMP expressions (Theorem 1, Fig. 1) on the x-axis of these test error plots (as it is valid even for overparameterized models). However, since MDL-COMP remains monotone with dimension d, the qualitative conclusion about the double descent of the test error does not change. Next, to investigate (ii) (estimators), we note from Fig. 3F that the tuned ridge estimators, either using cross-validation, or Prac-MDL-COMP, do exhibit the classical U-shaped curve with either choice of complexity measures, and obtain minimum test error at the true dimensionality of the data generating model. Furthermore, this conclusion is robust across ill-conditioned designs: Fig. F1 shows that tuned ridge estimators exhibit U-shaped test error curves even when the covariate matrix is ill-conditioned, and achieve superior test error than OLS. In fact, for our choice of covariate matrices, the OLS estimator exhibits double descent and *even multiple descent* in both low and high-dimensional settings, including non-overparameterized regimes.[16] Moreover, when using the best prediction error to do model selection (across the choice of number of features), the OLS estimator admits worse (sometimes significant worse) prediction error than its Ridge counterpart.

Related work on double descent: Recently, there has been a lot of research interest in probing the double descent phenomenon via different lenses. On the one hand, several works establish sufficient conditions for the OLS estimator to achieve good generalization in an overparameterized setting ($d \gg n$), a phenomenon referred to as the *benign overfitting* (Bartlett et al., 2020; Muthukumar et al., 2020; Hastie et al., 2019; Tsigler and Bartlett, 2020). It is worth noting that the OLS estimator can continue to exhibit double descent even for these settings. On the other hand, many works investigate the test error curves of ridge estimators for various settings: (a) When the dimensionality of the generative model d varies along with the fitted model (keeping the fitted model correctly specified), Hastie et al. (2019) prove that optimally tuned ridge estimators exhibit a U-shape curve for linear models and isotropic features. In their set-up, one plots the test error curve for range of d keeping the sample size fixed. (b) Nakkiran et al. (2020) make a similar conclusion for linear models with isotropic features but now for a fixed d while varying the sample size n. We highlight that both (a) and (b) are subtly different than the practical set-up discussed in the previous paragraph where the training observations are fixed and only the fitted models vary. In contrast, both (a) and (b) vary the training observations simultaneously with the

[15] Applied papers use fixed test set, but theory papers use varying training and test sets. Refer to the discussion on related work in the sequel.

[16] Here low and high-dimensional setting respectively denotes whether or not the true generative model has more parameters than the training sample size.

fitted estimator (and the fitted model always remains correctly specified). However, the qualitative conclusions drawn remain similar—tuned regularization restores the classical U-shape of test error curves.

Conclusions about the possible cause of double descent: Combining our findings along with the concurrent work, we can hypothesize that while parameter count is not a justified complexity measure, the double descent phenomenon is likely to be a consequence of a poor choice of estimators in ill-posed regimes. Poor estimators like OLS can exhibit double descent or even multiple descent depending on the covariate matrix, while regularized estimators with tuned hyper-parameters continue to exhibit U-shaped test error curves.

6.2 Future directions

We believe that our work takes a useful step towards questioning the fundamental components that underlie the principle of bias-variance tradeoff, and provides a proof of concept for the value of revisiting the complexity measure in overparameterized settings. Besides, several direct future directions arise from our work. Relating MDL-COMP with out-of-sample guarantees under additional assumptions on the covariate design, like those in (Hsu et al., 2012; Dobriban and Wager, 2018) for the linear model is an interesting open problem. Our results for kernel methods are in the classical low-dimensional regime, and it is known that several kernels in the high-dimensional regime behave very similar to the linear kernel (El Karoui, 2010a,b). Understanding the consequences in such high-dimensional regime for kernel methods with the MDL lens is another interesting future direction. Next, we note that our measures are based on ridge estimators but they are often not suitable for parameter estimation with sparse models in high-dimensions, and thus deriving MDL-COMP with codes suitably adapted for ℓ_1-regularization would also be interesting. Additionally, it remains to investigate suitable variants of our MDL–COMP or the more general one-part universal coding (Grünwald and Roos, 2019; Grünwald and Mehta, 2019) beyond linear and kernel methods, e.g., for deep neural networks, as well as for classification tasks.

Acknowledgments

This work was partially supported by National Science Foundation grant NSF-DMS-1613002, NSF-2015341, and the Center for Science of Information (CSoI), a US NSF Science and Technology Center, under grant agreement CCF-0939370, the NSF grant 2023505 on Collaborative Research: Foundations of Data Science Institute (FODSI), the NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning through awards DMS-2031883 and 814639, and an Amazon Research Award to BY, and Office of Naval Research grant DOD ONR-N00014-18-1-2640 and National Science Foundation grant NSF-DMS-2015454 to MJW.

Contents

Appendix A. Further discussion of MDL–COMP

We start with additional details on the simulation set-up and sketch related to the MDL–COMP scaling from Sec. 4.1 in Appendix A.1. In Appendix A.2, we sketch the proof for the MDL–COMP's behavior as observed in Fig. 1, and then provide an analytical bound for $\mathcal{R}_{\text{opt}}$ in Appendix A.3 under high-dimensional asymptotics. In Appendix A.4, we argue that the minimax optimality of MDL-COMP holds under a broad class of noise distributions (beyond Gaussian) under the linear model (21). Finally, we collect some additional background related to MDL in Appendix A.6.

A.1 Simulation set-up

Here we elaborate the set-up associated with Figs. 1 and 2. The true dimensionality $d_\star$ denotes that the observations $\mathbf{y} = \widetilde{\mathbf{X}}\theta_\star + \xi$ depend only on first $d_\star$ covariates of the full matrix $\mathbf{X}_{\text{full}}$, Given a sample size n (fixed for a given dataset, fixed in a given plot), the matrix $\mathbf{X}_{\text{full}} \in \mathbb{R}^{n \times \overline{d}}$, and $\widetilde{\mathbf{X}} \in \mathbb{R}^{n \times d_\star}$ where $\overline{d}$ denotes the maximum number of covariates available for fitting the model (and in our plots can be computed by multiplying the sample size with the the maximum value of d/n denoted on the x-axis). When we vary d, we use $\mathbf{X} \in \mathbb{R}^{n \times d}$ (selecting first d columns of $\mathbf{X}_{\text{full}}$) for fitting the ridge model and computing the MDL–COMP.

Defining w, and clarifying footnote 10: In order to compute MDL–COMP (28a), we need to compute the eigenvalues ρ_i of $\mathbf{X}^\top\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes the first d columns of the full matrix $\mathbf{X}_{\text{full}}$ that are used for fitting the ridge model, and computing the corresponding encoding and MDL–COMP. Moreover, we need to compute the vector $\mathbf{w}$

defined equal to $\mathbf{U}^\top \theta_\star$ in equation (9). Note that $\mathbf{U}$ has size $d \times d$ and $\theta_\star$ is $d_\star$ dimensional. So when $d < d_\star$, the vector $\mathbf{w}$ is computed by restricting $\theta_\star$ to first d dimensions (in equation (9)), i.e., using $\widetilde{\theta}_\star = (\theta_\star)_{[1:d]}$ (the orthogonal projection of $\theta_\star$ on $\mathbf{X}^\top\mathbf{X}$) and define $\mathbf{w} = \mathbf{U}^\top \widetilde{\theta}_\star$. On the other hand, for $d > d_\star$, we simply extend the $\theta_\star$ by appending $d - d_\star$ 0's, i.e., $\widetilde{\theta}_\star = \begin{bmatrix} \theta_\star \\ \mathbf{0}_{d-d_\star} \end{bmatrix}$ and then set $\mathbf{w} = \mathbf{U}^\top \widetilde{\theta}_\star$.

A.2 Proof sketch for random isotropic designs

In this section, we provide a proof sketch to explain the behavior of MDL–COMP/ when applied to random isotropic designs, as plotted in Fig. 1. Suppose that the design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has entries drawn from iid from the Gaussian distribution $\mathcal{N}(0, 1/n)$; the $1/n$-variance serves to ensure that $\mathbf{X}$ has columns with expected squared norm equal to one. When $n \gg d$, standard random matrix theory guarantees that $\mathbf{X}^\top\mathbf{X} \approx \mathbf{I}_d$ and hence $\rho_i \approx 1$. For $d \gg n$, one can apply the same argument to the matrix $\mathbf{X}\mathbf{X}^\top$ to conclude that $\mathbf{X}\mathbf{X}^\top \approx \frac{d}{n}\mathbf{I}_n$; thus, the matrix $\mathbf{X}^\top\mathbf{X}$ has rank n with its non-zero eigenvalues roughly equal to d/n. And, from above, we recall the definition $\mathbf{w} = \mathbf{U}^\top\widetilde{\theta}_\star$, where $\widetilde{\theta}_\star$ is either a restriction of $\theta_\star$ when the model is under-specified, or is obtained by appending zeros when the model is over-specified.

Since the matrix $\mathbf{U}$ consists of the eigenvectors of $\mathbf{X}^\top\mathbf{X}$, it has a uniform distribution over the space of all orthogonal matrices in dimension d. Let $r^2 := \|\theta_\star\|^2$ and, note from above that $\|\mathbf{w}\|^2 = \|\widetilde{\theta}_\star\|^2$. When $d \geq d_\star$, we have $\|\widetilde{\theta}_\star\|^2 = r^2$, and when $d < d_\star$, and the coordinates of $\theta_\star$ are drawn iid, we have $\|\widetilde{\theta}_\star\|^2 \approx \frac{d}{d_\star} r^2$. Given the distribution of $\mathbf{U}$, we have that the entries of w_i^2 are approximately equal, and thus conclude that $w_i^2 \approx \min\left\{1, \frac{d}{d_\star}\right\} \frac{r^2}{d} = r^2 \cdot \min\left\{\frac{1}{d}, \frac{1}{d_\star}\right\}$. Plugging in these approximations, when $d_\star < n$, we find that

$$\text{MDL–COMP} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log\left(\rho_i + \frac{\sigma^2}{w_i^2}\right) \approx \begin{cases} \dfrac{d}{2n} \log\left(1 + d_\star/r^2\right) & \text{if } d \in [1, d_\star] \\ \dfrac{d}{2n} \log\left(1 + d/r^2\right) & \text{if } d \in [d_\star, n] \\ \dfrac{1}{2} \log\left[d\left(\dfrac{1}{n} + \dfrac{1}{r^2}\right)\right] & \text{if } d \in [n, \infty) \end{cases} \tag{39}$$

$$\mathcal{R}_{\text{opt}} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i w_i^2}{\sigma^2}\right) \approx \begin{cases} \dfrac{d}{2n} \log\left(1 + \dfrac{r^2}{d_\star}\right) & \text{if } d \in [1, d_\star] \\ \dfrac{d}{2n} \log\left(1 + \dfrac{r^2}{d}\right) & \text{if } d \in [d_\star, n] \\ \dfrac{1}{2} \log\left(1 + \dfrac{r^2}{n}\right) & \text{if } d \in [n, \infty). \end{cases} \tag{40}$$

For the case when $d_\star > n$, i.e., the true dimensionality is larger than the sample size, we have

$$\text{MDL–COMP} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log\left(\rho_i + \frac{\sigma^2}{w_i^2}\right) \approx \begin{cases} \frac{d}{2n} \log\left(1 + \frac{d_\star}{r^2}\right) & \text{if } d \in [1, n] \\ \frac{1}{2} \log\left(\frac{d}{n} + \frac{d_\star}{r^2}\right) & \text{if } d \in [n, d_\star] \\ \frac{1}{2} \log\left[d\left(\frac{1}{n} + \frac{1}{r^2}\right)\right] & \text{if } d \in [d_\star, \infty), \end{cases} \tag{41}$$

$$\mathcal{R}_{\text{opt}} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i w_i^2}{\sigma^2}\right) \approx \begin{cases} \frac{d}{2n} \log\left(1 + \frac{r^2}{d_\star}\right) & \text{if } d \in [1, d_\star] \\ \frac{d}{2n} \log\left(1 + \frac{r^2}{d}\right) & \text{if } d \in [d_\star, n] \\ \frac{1}{2} \log\left(1 + \frac{r^2}{n}\right) & \text{if } d \in [n, \infty). \end{cases} \tag{42}$$

In both of the cases covered by equations (39) and (41), we find that the MDL-COMP has scaling $\mathcal{O}(d/n)$ for small d, and $\mathcal{O}(\log d)$ for $d \gg n$. Overall, this argument along with results in Fig. 1 suggest that d/n should not be treated as the default complexity for $d > n$, and that MDL–COMP provides a scaling of order $\log d$ for $d > n$.

A.3 Expressions for $\mathcal{R}_{\text{opt}}$ for large scale settings

In recent work, a number of authors (Hastie et al., 2019; Belkin et al., 2019b) have studied the generalization error of different estimators applied to linear models under the high-dimensional asymptotic scaling $d, n \to \infty$ and $d/n \to \gamma$. In a similar setting, we can derive an expression for the optimal redundancy $\mathcal{R}_{\text{opt}}$.

Theorem 4 *Suppose that the covariate matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has i.i.d. entries drawn from a distribution with mean 0, variance $1/n$ and fourth moments of order $1/n^2$, and that the parameter $\theta_\star$ is drawn randomly from a rotationally invariant distribution, with $\mathbb{E}[\frac{\|\theta_\star\|^2}{\sigma^2 d}] = \text{snr}$. Then, we have*

$$\lim_{n,d \to \infty, \frac{d}{n} \to \gamma} \mathbb{E}_{\theta_\star}[\mathcal{R}_{opt}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}})] \leq \gamma \log(1 + \text{snr} - \delta) + \log(1 + \gamma \cdot \text{snr} - \delta) - \frac{\delta}{\text{snr}}, \tag{43}$$

almost surely, where $\delta := (\sqrt{\text{snr}(1+\sqrt{\gamma})^2 + 1} - \sqrt{\text{snr}(1-\sqrt{\gamma})^2 + 1})^2/4$.

See Appendix C.4 for the proof. We remark that the inequality in the theorem is a consequence of Jensen's inequality and can be removed whenever a good control over the quantity $\mathbb{E}_{w_i^2} \log(1 + w_i^2 \rho_i)$ is available, where ρ_i denoting the eigenvalues of $\mathbf{X}^\top \mathbf{X}$. The first term on the RHS of equation (43) has a scaling of order d/n when the norm of $\theta_\star$ grows with d. However, when $\|\theta_\star\|$ is held constant with d, the optimal redundancy $\mathcal{R}_{\text{opt}}$ does not grow and remains bounded by a constant. The bound (43) has a similar scaling as that noted earlier in equations (40) and (42) from our proof sketch (for $d \in [d_\star, n]$ or $d \in [n, \infty)$, assuming $d_\star$ fixed, r^2 growing with d).

A.4 MDL–COMP over a wider range of noise distributions

Theorem 1 provides an explicit expression for MDL–COMP assuming that the noise follows a Gaussian distribution. In this section, we show that the optimal code from equation (22b) defining the MDL–COMP procedure (22a) also achieves a minimax codelength over a wider range of noise distributions. Let $\mathcal{P}$ denote the set of all distributions on $\mathbb{R}^n$, and define the family

$$\mathcal{P}_{\text{bndvar}} = \left\{ \mathbb{P} \in \mathcal{P} : \mathbb{E}[\xi] = 0, \text{Var}(\xi) \preceq \sigma^2 \mathbf{I}_n \right\}. \tag{44}$$

For the generative linear model (21) with noise distribution now allowed to belong to the family $\mathcal{P}_{\text{bndvar}}$ (44), we have

$$\mathbb{E}_\xi[\mathbf{y}|\mathbf{X}] = \mathbf{X}\theta_\star, \quad \text{and} \quad \text{Var}_\xi(\mathbf{y}|\mathbf{X}) \preceq \sigma^2 \mathbf{I}_n.$$

Our next result shows that the code defininining MDL–COMP also achieves the minimax codelength over the noise distributions in $\mathcal{P}_{\text{bndvar}}$.

Theorem 5 *The distribution $\mathbb{Q}_{\mathbf{\Lambda}_{opt}}$ that achieves the* MDL–COMP *in equation* (22a) *also achieves the minimax codelength in the class $\mathcal{P}_{bndvar}$, i.e.,*

$$\mathbb{Q}_{\mathbf{\Lambda}_{opt}} \in \arg \min_{\mathbb{Q} \in \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}} \max_{\mathbb{P} \in \mathcal{P}_{bndvar}} \mathbb{E}_{\xi \sim \mathbb{P}} \log \left(\frac{1}{q(\mathbf{y})} \right), \tag{45}$$

where q denotes the density of the distribution $\mathbb{Q}$.

See Appendix C.5 for the proof of this claim.

A.5 MDL-COMP with multiplicative factor 1 in equation (17)

Note that we change the multiplicative factor of the codelength for $\mathbf{\Lambda}$ in equation (17) from $\frac{1}{2}$ to 1, and thereby the definition of MDL–COMP in equation (17) the expression (28a) for MDL-COMP in Theorem 1 becomes

$$\text{MDL–COMP}(\mathbb{P}_{\theta_\star}, \mathcal{Q}^{\mathbf{X}}_{\text{Ridge}}) = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log \left(\frac{\rho_i \sigma^2}{w_i^2} + \frac{\sigma^4}{w_i^4} \right),$$

where we use the following results from the proof of Theorem 1:

$$\lambda_i^{\text{opt}} = \frac{\sigma^2}{w_i^2} \quad \text{and} \quad \mathcal{R}_{\text{opt}} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log \left(1 + \frac{\rho_i}{\lambda_i^{\text{opt}}} \right),$$

which then yields the following expressions in place of equation (39) for the approximate scaling of MDL–COMP for random isotropic designs:

$$\text{MDL–COMP} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log \left(\frac{\rho_i \sigma^2}{w_i^2} + \frac{\sigma^4}{w_i^4} \right) \approx \begin{cases} \frac{d}{2n} \left[\log \left(1 + d_\star / r^2 \right) + \log(d_\star / r^2) \right] & \text{if } d \in [1, d_\star] \\ \frac{d}{2n} \left[\log \left(1 + d / r^2 \right) + \log(d / r^2) \right] & \text{if } d \in [d_\star, n] \\ \frac{1}{2} \log \left[d \left(\frac{1}{n} + \frac{1}{r^2} \right) \right] & \text{if } d \in [n, \infty), \end{cases}$$

and we can bound all of these. Analogous expressions in place of equation (41) can be derived similarly. In all cases, the new expressions can be bounded by at most 2 times of the previous expressions, so that the scaling of MDL–COMP remains unaffected.

A.6 Relation between NML and maximum likelihood principle

As noted earlier, for computing the optimal redundancy in equation (7), one can replace the set of codes $\mathcal{Q}_{\text{Ridge}}$ by a generic set of codes $\mathcal{Q}$ corresponding to many classes of models. In contrast, for maximum likelihood estimation (MLE), one maximizes the log likelihood of the data over a fixed model class. Maximum likelihood is generally used when a model class is fixed, and is known to break down when considering even nested classes of parametric models (Hansen and Yu, 2001). On the other hand, the definition (7) can be suitably adjusted even for the case when there is no true generative model for $\mathbf{y}$. At least in general, the quantity $\mathbb{Q}(\mathbf{y})$ need not denote the likelihood of the observation $\mathbf{y}$, and the distribution $\mathbb{Q}$ may not even correspond to a generative model. In such cases, the optimal choice of $\mathbb{Q}$ in equation (7) is supposed to be optimal not just for $\mathbf{y}$ from a parametric family, but also for $\mathbf{y}$ from one of many parametric model classes of different dimensions.

However, when $\mathcal{Q}$ is a single parametric family, i.e., $\mathcal{Q} = \{\mathbb{P}_\theta, \theta \in \Theta\}$ where θ denotes the unknown parameter of interest, the MDL principle does reduce to the MLE principle. In more precise terms, the MLE can be seen as the continuum limit of two-part MDL (see Chapter 15.4 (Grünwald, 2007)). In this case, the optimal NML code $\mathbb{Q}^\star(\mathbf{y})$ is given by the logarithm of the likelihood computed at MLE plus a term that depends on the normalization constant of the maximum likelihood over all possible observations; this fact underlies the nomenclature of normalized maximum likelihood, or NML for short.

For the low-dimensional linear models (fixed d and $n \to \infty$), while several MDL-based complexities, namely two-part codelength, stochastic information complexity, and NML complexity are equivalent to the first $\mathcal{O}(\log n)$ term—which in turn scales like $\frac{1}{2} d \log n$. Moreover, the NML complexity can be deemed optimal when accounting for the constant term, i.e., NML complexity achieves the optimal $\mathcal{O}(1)$ term which involves Jeffrey's prior (Barron et al., 1998), asymptotically for low-dimensional linear models. We refer readers to the sources (Rissanen, 1986; Barron et al., 1998; Hansen and Yu, 2001) for further discussion on two-part coding, stochastic information complexity, and toSections 5.6, 15.4 of the book (Grünwald, 2007) for further discussion on the distinctions between MDL and MLE.

Appendix B. Further numerical experiments

We now present additional experiments showing the usefulness of the data-driven Prac-MDL-COMP (34). In particular, we show that Prac-MDL-COMP informs out-of-sample MSE in Gaussian as well as several misspecified linear models (Appendix B.1), and then provide further insight on the performance of real-data experiments (deferred from Sec. 5.3 of the main paper) in Appendix B.2.

B.1 Misspecified linear models

We specify three different types of model misspecification taken from prior work (Yu and Kumbier, 2020) and analyze the ability of MDL-COMP to select models that generalize.

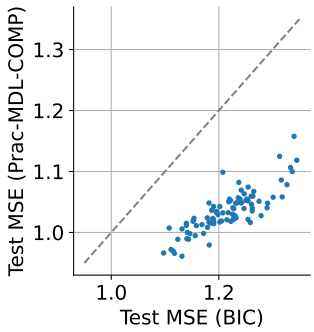

Figure B1. Prac-MDL-COMP successfully selects models which predict fMRI responses well compared to the BIC criterion (comparisons with cross-validation and Ridge-ARD shown in Fig. 5).

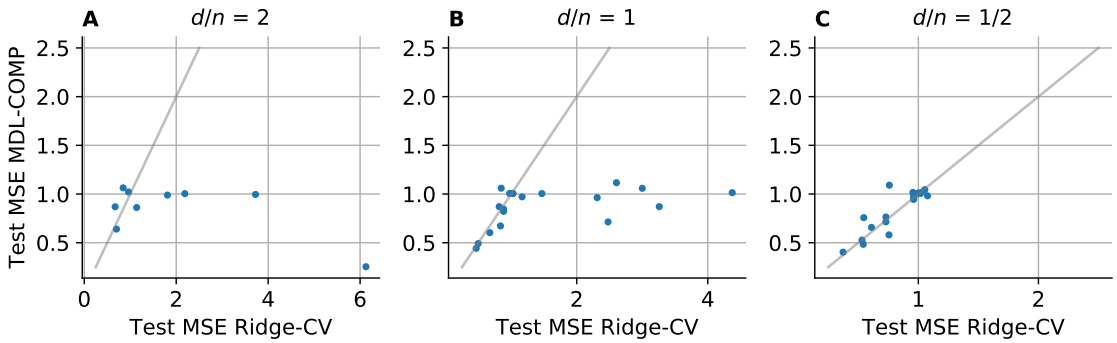

Figure B2: Results for Fig 4 hold when using 5-fold CV rather than LOOCV.

Fig. B3 shows that under these conditions, MDL-COMP still manages to pick a λ which generalizes fairly well. *T-distribution* refers to errors being distributed with a t-distribution with three degrees of freedom. *Decaying eigenvalues* refers to the eigenvalues of the covariance matrix λ_i decaying as $1/2^i$, inducing structure in the covariance matrix. *Thresholds* refers to calculating the outcome using indicator functions for $X > 0$ in place of X. Here, Ridge-CV (orange dotted line) refers to model selection using leave-one-out cross-validation and Prac-MDL-COMP (blue line) refers to model selection for a ridge model based on optimizing Prac-MDL-COMP.

B.2 Real data experiments continued

We now provide further investigation. First, the good performance of Prac-MDL-COMP based linear model also holds ground against 5-fold CV, see Fig. B2.

Here we show results for 28 real datasets in Fig. B4 where the plot titles correspond to dataset IDs in OpenML (Vanschoren et al., 2013). In the limited data regime (when d/n is large, the right-hand side of each plot), MDL-COMP tends to outperform Ridge-CV. As the number of training samples is increased, the gap closes or cross-validation

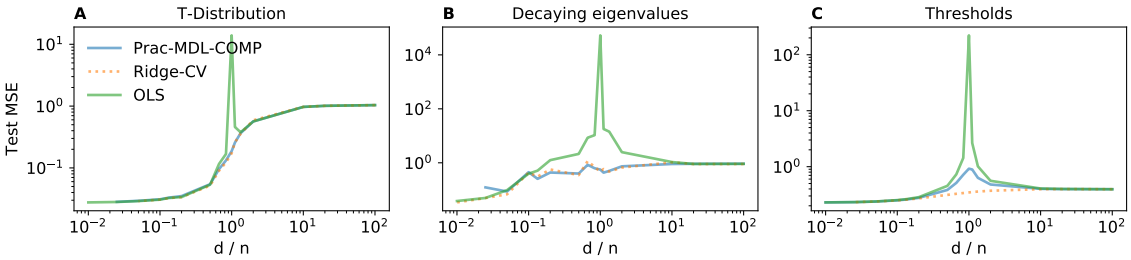

Figure B3. Model selection under misspecification. Under various misspecifications, Prac-MDL-COMP still manages to select models which generalize reasonably well. Different from other figures presented in the paper, here d is fixed and the sample size n is varied.

begins to outperform MDL-COMP. These observations provide evidence for promises of Prac-MDL-COMP based regularization for limited data settings.

Appendix C. Proofs

This appendix serves to collect the proofs of all of our results. The proofs for Theorems 1 through 5 are provided in Appendix C.1 through Appendix C.5, and for Corollary 1 in Appendix C.6.

C.1 Proof of Theorem 1

Our proof is based on establishing that

$$\lambda_i^{\text{opt}} = \frac{\sigma^2}{w_i^2} \quad \text{and} \quad \mathcal{R}_{\text{opt}} = \frac{1}{2n} \sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i}{\lambda_i^{\text{opt}}}\right). \tag{46}$$

The claimed expressions for $\mathcal{R}_{\text{opt}}$ and MDL–COMP follow by using these expressions and performing some algebra.

Let $\mathbb{P}_{\theta_\star}$ denote the distribution of the multivariate Gaussian $\mathcal{N}(\mathbf{X}\theta_\star, \sigma^2\mathbb{I}_n)$, and let $p(\mathbf{y}; \mathbf{X}, \theta_\star)$ denote its density. Note $\mathbf{y} \sim \mathbb{P}_{\theta_\star}$ by assumption.[17] In order to simplify notation, we introduce the shorthand $\widehat{\theta} = \widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y})$.

[17] In our our earlier discussion with under-specified models in the simulations related to Fig. 1, and the discussion in Appendices A.1 and A.2, $\mathbf{X}$ denotes a subset of full features, in which case $\mathbf{X}\theta_\star$ does not capture the mean of the random vector $\mathbf{y}$, and there is a bias. However, given the definition of our MDL-COMP, where we compare to the best possible encoding using $\mathbf{X}$ at hand, this bias term arises in both the numerator and denominator in equation (47), and cancels out thereby not affecting the subsequent derivations. On the other hand, with over-specified models, i.e., when $\mathbf{X}$ is a superset of features needed to correctly specify the mean of the random vector $\mathbf{y}$, we can append zeros to the true parameter $\theta_\star$ as necessary, and continue to assume $\mathbf{y} \sim \mathbb{P}_{\theta_\star}$.

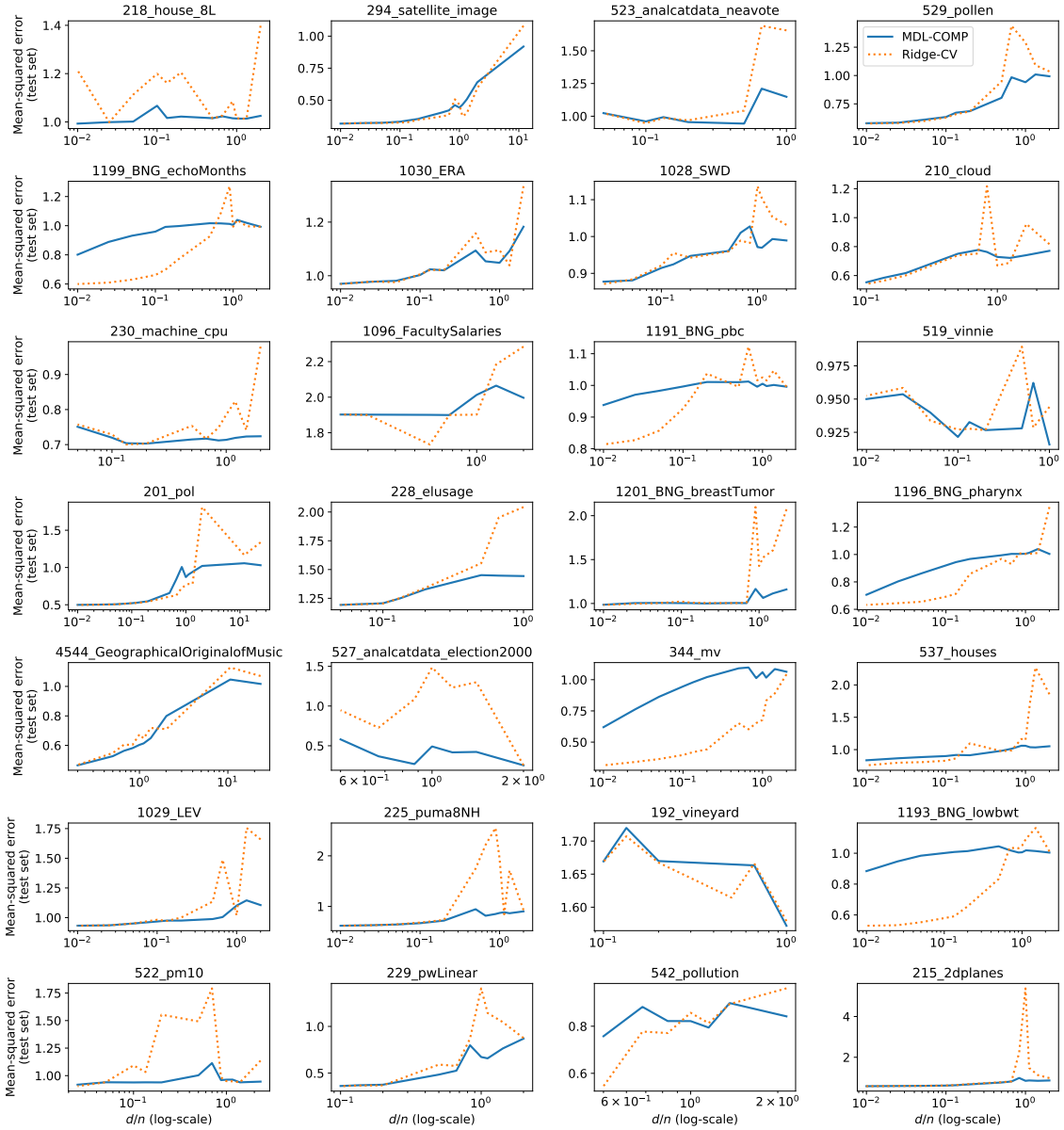

Figure B4. Test MSE when varying the number of training points sampled from real datasets (lower is better). MDL-COMP performs well, particularly when the number of training points is small (right-hand side of each plot). Each point averaged over 3 random bootstrap samples.

We have

$$\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{\theta_\star} \,\|\, \mathbb{Q}_{\mathbf{\Lambda}}) = \mathbb{E}_{\mathbf{y}}\left[\log\left(\frac{\frac{1}{(2\pi\sigma^2)^{n/2}}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{X}\theta_\star\|^2\right)}{\frac{1}{\mathrm{C}_{\mathbf{\Lambda}}(2\pi\sigma^2)^{n/2}}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{X}\widehat{\theta}\|^2 - \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta}\right)}\right)\right] \tag{47}$$

$$= \sum_{j=1}^{3} T_j,$$

Dataset name (OpenML ID)	#obs n	#feats d	Ridge-CV MSE	Prac-MDL-COMP MSE
1028_SWD	1000	11	1.17	**0.97**
1029_LEV	1000	5	NaN*	**1.10**
1030_ERA	1000	5	NaN*	**1.05**
1096_FacultySalaries	50	5	NaN*	**2.01**
1191_BNG_pbc	1000000	19	1.02	**1.00**
1193_BNG_lowbwt	31104	10	**0.98**	1.01
1196_BNG_pharynx	1000000	11	1.03	**1.00**
1199_BNG_echoMonths	17496	10	1.47	**1.00**
1201_BNG_breastTumor	116640	10	2.61	**1.12**
192_vineyard	52	3	NaN*	**1.57**
201_pol	15000	49	**0.82**	0.87
210_cloud	108	6	3.41	**0.73**
215_2dplanes	40768	11	0.88	**0.85**
218_house_8L	22784	9	4.37	**1.01**
225_puma8NH	8192	9	3.26	**0.87**
228_elusage	55	3	NaN*	**1.44**
229_pwLinear	200	11	0.83	**0.67**
230_machine_cpu	209	7	2.48	**0.71**
294_satellite_image	6435	37	0.47	**0.44**
344_mv	40768	11	**0.85**	1.06
4544_GeographicalOriginalofMusic	1059	118	0.67	**0.60**
519_vinnie	380	3	NaN*	**0.92**
522_pm10	500	8	2.31	**0.96**
523_analcatdata_neavote	100	3	NaN*	**1.15**
527_analcatdata_election2000	67	15	0.50	**0.49**
529_pollen	3848	5	NaN*	**0.94**
537_houses	20640	9	3.00	**1.06**
542_pollution	60	16	0.88	**0.82**

Table B1. Prac-MDL-COMP vs Cross-validation for a range of datasets when the training data is limited. This table contains details about the datasets used in Figs. 4, B2 and B4. The first column denotes the name of the datasets, and the second and third columns report the size of the datasets. In the last two columns, we report the performance for CV-tuned Ridge (LOOCV), and Prac-MDL-COMP tuned ridge estimator, trained on a subsampled dataset such that $d/n = 1$. The reported MSE is computed on a hold-out testing set (which consists of 25% of the observations), and the better (of the two) MSE for each dataset is highlighted in bold. We observe that Prac-MDL-COMP based tuning leads to superior performance compared to cross-validation for most of the datasets. *When too few points are available, cross-validation fails to numerically fit for low values of λ, returning a value of NaN.

where

$$T_1 := -\mathbb{E}_{\mathbf{y}}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\theta_\star\|^2\right], \quad T_2 := \mathbb{E}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top \mathbf{\Lambda}\widehat{\theta}\right], \tag{48}$$

and $T_3 := \log \mathrm{C}_{\mathbf{\Lambda}}$. By inspection, we have $T_1 = -n/2$. Dividing both sides by n yields

$$\mathcal{R}_{\text{opt}} = \min_{\mathbf{\Lambda}\in\mathcal{M}}\left\{\mathbb{E}\left[\frac{\|\mathbf{y} - \mathbf{X}\widehat{\theta}\|^2 + \widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta}}{2n\sigma^2}\right] + \frac{1}{2n}\sum_{i=1}^{d}\log\left(1 + \frac{\rho_i}{\lambda_i}\right)\right\} - \frac{1}{2} \tag{49}$$

Next we claim that

$$T_2 = \frac{(n - \min\{n,d\})}{2} + \frac{1}{2}\sum_{i=1}^{\min\{n,d\}} \frac{(\rho_i w_i^2/\sigma^2 + 1)\lambda_i}{\lambda_i + \rho_i}, \quad \text{and} \tag{50a}$$

$$T_3 = \log \mathrm{C}_{\mathbf{\Lambda}} = \frac{1}{2}\sum_{i=1}^{\min\{n,d\}} \log\left(\frac{\rho_i + \lambda_i}{\lambda_i}\right) \tag{50b}$$

Assuming these claims as given at the moment, let us now complete the proof. We have

$$\frac{1}{n}\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{\theta_\star} \,\|\, \mathbb{Q}_{\mathbf{\Lambda}}) = \frac{1}{n}\sum_{j=1}^{3} T_j = -\frac{\min\{n,d\}}{2n} + \frac{1}{2n}\sum_{i=1}^{\min\{n,d\}} f_i(\lambda_i), \qquad \text{where} \tag{51a}$$

$$f_i(\lambda_i) := \left(\frac{(\rho_i w_i^2/\sigma^2 + 1)\lambda_i}{\lambda_i + \rho_i} + \log\left(\frac{\rho_i + \lambda_i}{\lambda_i}\right)\right). \tag{51b}$$

Finally, in order to compute $\mathcal{R}_{\mathrm{opt}}$ defined in equation (49), we need to minimize the KL-divergence (51) as a function of the vector $\lambda = (\lambda_1, \ldots, \lambda_{\min\{n,d\}})$. Note that the objective on the RHS of equation (51) is separable across the components of λ, so we need only solve a univariate problem for each λ_i. Taking derivatives to find stationary points, we have

$$f_i'(\lambda_i) = 0 \quad \Longleftrightarrow \quad -\frac{(\rho_i w_i^2/\sigma^2 + 1)}{(1 + \rho_i/\lambda_i)^2} + \frac{1}{1 + \rho_i/\lambda_i} = 0 \quad \Longleftrightarrow \quad \lambda_i^{\mathrm{opt}} = \frac{\sigma^2}{w_i^2}. \tag{52}$$

Moreover, checking on the boundary of feasible values of $\lambda_i = [0, \infty)$, we have $f_i(\lambda_i) \to \infty$ as $\lambda_i \to 0^+$, and $f_i(\lambda_i) \to 1 + \rho_i w_i^2/\sigma^2$ as $\lambda_i \to \infty$. Noting that $1 + \log a \le a$ for all $a \ge 1$, we conclude that f_i achieves its minimum at λ_i^{opt}, and we have $f_i(\lambda_i^{\mathrm{opt}}) = 1 + \log(1 + \rho_i w_i^2/\sigma^2)$. Substituting this value into the expression (52) yields

$$\begin{aligned}\mathcal{R}_{\mathrm{opt}} &= -\frac{\min\{n,d\}}{2n} + \frac{1}{2n}\sum_{i=1}^{\min\{n,d\}} \left(1 + \log\left(1 + \frac{\rho_i w_i^2}{\sigma^2}\right)\right) \\ &= \frac{1}{2n}\sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i w_i^2}{\sigma^2}\right).\end{aligned} \tag{53}$$

We now turn to proving our earlier claims (50a) and (50b). We prove them separately. We start with the low-dimensional case, i.e., when $d < n$. Since the proof for the high-dimensional case is similar, we only outline the mains steps in Appendix C.1.4.

C.1.1 Proof of claims (50a) and (50b) for the case $d < n$

Let the singular value decomposition of the matrix $\mathbf{X}$ be given by

$$\mathbf{X} = \mathbf{V}\sqrt{\mathbf{D}}\mathbf{U}^\top. \tag{54}$$

Here the matrix $\mathbf{D} \in \mathbb{R}^{d \times d}$ is a diagonal matrix, with i-th diagonal entry denoting the i-th squared singular value of the matrix $\mathbf{X}$. Moreover, the matrices $\mathbf{V} \in \mathbb{R}^{n \times d}$ and $\mathbf{U} \in \mathbb{R}^{d \times d}$,

respectively,contain the left and right singular vectors of the matrix $\mathbf{X}$, respectively. (I.e., the i-th column denotes the singular vector corresponding the i-th singular value.) Note that we have the relations $\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_d$, and moreover, the matrix $\mathbf{V}\mathbf{V}^\top$ is a projection matrix of rank d. Finally, let $\overline{\mathbf{\Lambda}} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ be such that $\mathbf{\Lambda} = \mathbf{U}\overline{\mathbf{\Lambda}}\mathbf{U}^\top$. With this notation in place, we claim that

$$(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top)^2 = \mathbf{V}\overline{\mathbf{\Lambda}}^2(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-2}\mathbf{V}^\top + (\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top) \tag{55a}$$

$$\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{\Lambda}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top = \mathbf{V}\mathbf{D}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-2}\mathbf{V}^\top, \quad \text{and} \tag{55b}$$

$$\begin{aligned}\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\mathbf{X}\theta_\star &= \mathbf{V}\sqrt{\mathbf{D}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{D}\mathbf{U}^\top\theta_\star \\ &= \mathbf{V}\sqrt{\mathbf{D}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{D}\mathbf{w}\end{aligned} \tag{55c}$$

See Appendix C.1.5 for the proofs of these claims, which involve elementary linear algebra.

C.1.2 Proof of the expression (50a) for term T_2

We have

$$\begin{aligned}&\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta} \\ &= \frac{1}{2\sigma^2}\| \left(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\right)\mathbf{y})\|^2 + \frac{1}{2\sigma^2}\|\sqrt{\mathbf{\Lambda}}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\mathbf{y}\|^2 \\ &\overset{(i)}{=} \frac{1}{2\sigma^2}\mathbf{y}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}^2(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-2}\mathbf{V}^\top\right)\mathbf{y} + \frac{1}{2\sigma^2}\mathbf{y}^\top(\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top)\mathbf{y} + \frac{1}{2\sigma^2}\mathbf{y}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}\mathbf{D}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-2}\mathbf{V}^\top\right)\mathbf{y} \\ &= \frac{1}{2\sigma^2}\mathbf{y}^\top\left(\mathbf{V}(\overline{\mathbf{\Lambda}}^2 + \overline{\mathbf{\Lambda}}\mathbf{D})(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-2}\mathbf{V}^\top\right)\mathbf{y} + \frac{1}{2\sigma^2}\mathbf{y}^\top(\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top)\mathbf{y} \\ &= \frac{1}{2\sigma^2}\mathbf{y}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right)\mathbf{y} + \frac{1}{2\sigma^2}\mathbf{y}^\top(\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top)\mathbf{y},\end{aligned} \tag{56}$$

where step (i) follows from equations (55a) and (55b). The latter steps make use of the fact that diagonal matrices commute.

Next we note that $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\theta_\star, \sigma^2\mathbb{I}_n)$, and thus $\mathbb{E}[\mathbf{y}^\top\mathbf{A}\mathbf{y}] = \theta_\star^\top\mathbf{X}^\top\mathbf{A}\mathbf{X}\theta_\star + \sigma^2\,\mathrm{trace}(\mathbf{A})$. Using equations (48) and (56), we find that

$$\begin{aligned}T_2 &= \mathbb{E}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta}\right] \\ &= \frac{1}{2\sigma^2}\theta_\star^\top\mathbf{X}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right)\mathbf{X}\theta_\star + \frac{1}{2\sigma^2}[\sigma^2\,\mathrm{trace}(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top)] \\ &\qquad + \frac{1}{2\sigma^2}\theta_\star^\top\mathbf{X}^\top(\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbf{X}\theta_\star + \frac{1}{2\sigma^2}[\sigma^2\,\mathrm{trace}(\mathbf{I} - \mathbf{V}\mathbf{V}^\top)] \\ &\overset{(i)}{=} \frac{1}{2\sigma^2}\theta_\star^\top\mathbf{X}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right)\mathbf{X}\theta_\star + \frac{1}{2\sigma^2}[\sigma^2\,\mathrm{trace}(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top)] \\ &\qquad + 0 + \frac{1}{2}\cdot(n - d) \\ &\overset{(ii)}{=} \frac{1}{2\sigma^2}\left(\mathbf{w}^\top\mathbf{D}\mathbf{\Lambda}(\mathbf{D} + \mathbf{\Lambda})^{-1}\mathbf{w} + \sigma^2\,\mathrm{trace}[\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}]\right) + \frac{1}{2}\cdot(n - d) \\ &= \frac{(n - d)}{2} + \frac{1}{2}\sum_{i=1}^{d}\left(\frac{\rho_i w_i^2}{\sigma^2} + 1\right)\frac{\lambda_i}{\lambda_i + \rho_i}\end{aligned} \tag{57}$$

$$\tag{58}$$

where step (i) follows from the facts that the matrix $(\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top)$ is a projection matrix of rank $n-d$, and is orthogonal to the matrix $\mathbf{X}$, i.e., $(\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top)\mathbf{X} = 0$. Step (ii) follows from a similar computation as that done to obtain equation (56), along with claim (55c), and the following identity for the matrix trace

$$\mathrm{trace}(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top) = \mathrm{trace}(\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\mathbf{V}) = \mathrm{trace}(\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{I}_d).$$

C.1.3 Proof of claim (50b) (term T_3):

From the normalization of a multivariate Gaussian density, we have

$$\frac{1}{(2\pi\sigma^2)^{n/2}}\int_{\mathbf{y}\in\mathbb{R}^n}\exp\left(-\frac{\mathbf{y}^\top\mathbf{A}\mathbf{y}}{2\sigma^2}\right)d\mathbf{y} = \sqrt{\det(\mathbf{A}^{-1})}, \tag{59}$$

valid for any positive definite matrix $\mathbf{A}\in\mathbb{R}^{n\times n}$.

Putting together the definition (14) of $\mathrm{C}_{\mathbf{\Lambda}}$ and equation (56), we find that

$$\mathrm{C}_{\mathbf{\Lambda}} = \frac{1}{(2\pi\sigma^2)^{n/2}}\int_{\mathbf{y}\in\mathbb{R}^n}\exp\left(-\frac{\mathbf{y}^\top\mathbf{A}_{\mathbf{\Lambda}}\mathbf{y}}{2\sigma^2}\right)d\mathbf{y} \quad \text{where} \quad \begin{aligned}\mathbf{A}_{\mathbf{\Lambda}} &= \left(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right) + (\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top)\\ &= \mathbf{I}_n - \mathbf{V}\mathbf{D}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top.\end{aligned} \tag{60}$$

The eigenvalues of the $n\times n$ matrix $\mathbf{A}_{\mathbf{\Lambda}}$ are given by $\left\{\frac{\lambda_1}{\lambda_1+\rho_1}, \frac{\lambda_2}{\lambda_2+\rho_2}, \ldots, \frac{\lambda_d}{\lambda_d+\rho_d}, 1, 1, \ldots, 1\right\}$, where the multiplicity of the eigenvalue 1 is $n-d$. Finally, applying equation (59), we find that

$$T_3 = \log \mathrm{C}_{\mathbf{\Lambda}} = \log\sqrt{\det(\mathbf{A}_{\mathbf{\Lambda}}^{-1})} = \frac{1}{2}\sum_{i=1}^{d}\log\left(\frac{\rho_i+\lambda_i}{\lambda_i}\right) \tag{61}$$

and the claim follows.

C.1.4 Proof of claims (50a) and (50b) for the case $d > n$

In this case, the dimensions of the matrices in the singular value decomposition (54) changes. The argument for the proof remains similar with suitable adaptations due to the change in the size of the matrices. As a result, we only outline the main steps.

We write

$$\mathbf{X} = \mathbf{V}\begin{bmatrix}\sqrt{\widetilde{\mathbf{D}}} & \mathbf{0}\end{bmatrix}\mathbf{U}^\top \quad\Longrightarrow\quad \mathbf{X}^\top\mathbf{X} = \mathbf{U}\underbrace{\begin{bmatrix}\widetilde{\mathbf{D}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}}_{\mathbf{D}}\mathbf{U}^\top \tag{62}$$

where $\mathbf{V}\in\mathbb{R}^{n\times n}$, $\widetilde{\mathbf{D}}\in\mathbb{R}^{n\times n}$ and $\mathbf{U}\in\mathbb{R}^{d\times d}$. Note that the non-zero entries of the matrix $\mathbf{D}$ are precisely the ones denoted by $\widetilde{\mathbf{D}}$.

Let $\overline{\mathbf{\Lambda}}_n = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ denote the $n \times n$ principal minor of the matrix $\overline{\mathbf{\Lambda}}$, where $\mathbf{\Lambda} = \mathbf{U}\overline{\mathbf{\Lambda}}\mathbf{U}^\top$. With these notations, we find that the claims (55) are replaced by

$$(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top)^2 = \mathbf{V}\overline{\mathbf{\Lambda}}_n^2(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-2}\mathbf{V}^\top \tag{63a}$$

$$\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{\Lambda}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top = \mathbf{V}\widetilde{\mathbf{D}}\overline{\mathbf{\Lambda}}_n(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-2}\mathbf{V}^\top \tag{63b}$$

$$\begin{aligned}\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\mathbf{X}\theta_\star &= \mathbf{V}\begin{bmatrix}\sqrt{\widetilde{\mathbf{D}}} & \mathbf{0}\end{bmatrix}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{D}\underbrace{\mathbf{U}^\top\theta_\star}_{=:\mathbf{w}} \\ &= \mathbf{V}\sqrt{\widetilde{\mathbf{D}}}(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-1}\widetilde{\mathbf{D}}\mathbf{w}_{1:n} \end{aligned} \tag{63c}$$

where $\mathbf{w}_{1:n}$ denotes the first n entries of the vector $\mathbf{w}$. The proof of these claims can be derived in a similar manner to that of claims (55) (see Appendix C.1.5).

Applying equations (63a) and (63b), we find that the equation (56) is modified to

$$\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta} = \frac{1}{2\sigma^2}\mathbf{y}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}_n(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-1}\mathbf{V}^\top\right)\mathbf{y}$$

which along with equation (63c) implies that the equation (58) is now replaced by

$$T_2 = \mathbb{E}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta}\right] = \frac{1}{2}\sum_{i=1}^{n}\left(\frac{\rho_i w_i^2}{\sigma^2} + 1\right)\frac{\lambda_i}{\lambda_i + \rho_i}.$$

Thus, we obtain the claimed expression (50a) for T_2 when $d > n$.

Next, to prove (50b) for this case, we find that the matrix $\mathbf{A}_{\mathbf{\Lambda}}$ (defined in equation (60)) for this case gets modified to

$$\mathbf{A}_{\mathbf{\Lambda}} = \mathbf{I}_n - \mathbf{V}\mathbf{D}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top.$$

Since $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_n$, we find that the eigenvalues of the $n \times n$ matrix $\mathbf{A}_{\mathbf{\Lambda}}$ are given by $\left\{\frac{\lambda_1}{\rho_1 + \lambda_1}, \ldots, \frac{\lambda_n}{\rho_n + \lambda_n}\right\}$. Therefore, we conclude that

$$T_3 = \log \mathrm{C}_{\mathbf{\Lambda}} = \log\sqrt{\det(\mathbf{A}_{\mathbf{\Lambda}}^{-1})} = \frac{1}{2}\sum_{i=1}^{n}\log\left(\frac{\rho_i + \lambda_i}{\lambda_i}\right).$$

C.1.5 Proof of claims (55a) and (63a)

For completeness, we discuss the proofs of the linear-algebra claims (55) and (63). Here we establish the claim (55a) (for $n > d$) and (63a) (for $n < d$). The other claims in the equations (55) and (63) can be derived in a similar fashion. Note that for both cases $n > d$ and $n < d$, our notations (54) and (62) are set-up such that

$$(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1} = \mathbf{U}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{U}^\top,$$

where the inverse is well-defined since $\overline{\mathbf{\Lambda}} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ is assumed to be a positive definite matrix. We use the fact that diagonal matrices commute, several times in our arguments.

Proof of claim (55a): Using the decomposition (54) for $n > d$ and noting that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_d$, we find that

$$\begin{aligned}(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top)^2 &= (\mathbf{I}_n - \mathbf{V}\sqrt{\mathbf{D}}\mathbf{U}^\top\mathbf{U}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{U}^\top\mathbf{U}\sqrt{\mathbf{D}}\mathbf{V}^\top)^2 \\ &= (\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top + \mathbf{V}\mathbf{V}^\top - \mathbf{V}\sqrt{\mathbf{D}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\sqrt{\mathbf{D}}\mathbf{V}^\top)^2 \\ &= (\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top + \mathbf{V}(\mathbf{I}_d - \sqrt{\mathbf{D}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\sqrt{\mathbf{D}})\mathbf{V}^\top)^2 \\ &= (\mathbf{I}_n - \mathbf{V}\mathbf{V}^\top + \mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top)^2 \\ &= \mathbf{I}_n - \mathbf{V}\mathbf{V}^\top + \mathbf{V}\overline{\mathbf{\Lambda}}^2(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-2}\mathbf{V}^\top,\end{aligned}$$

where the last step follows from the following facts (a) $\mathbf{I} - \mathbf{V}\mathbf{V}^\top$ is a projection matrix, (b) $(\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbf{V} = \mathbf{0}$, and (c) $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_d$.

Proof of claim (63a): Note that for the decomposition (62), we have $\mathbf{V}\mathbf{V}^\top = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_n$ and $\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}_d$. Using these facts along with the notation $\overline{\mathbf{\Lambda}}_n = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ for $d > n$, we find that

$$\begin{aligned}(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top)^2 &= (\mathbf{I}_n - \mathbf{V}\begin{bmatrix}\sqrt{\widetilde{\mathbf{D}}} & \mathbf{0}\end{bmatrix}\mathbf{U}^\top\mathbf{U}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{U}^\top\mathbf{U}\begin{bmatrix}\sqrt{\widetilde{\mathbf{D}}} \\ \mathbf{0}\end{bmatrix}\mathbf{V}^\top)^2 \\ &= (\mathbf{I}_n - \mathbf{V}\begin{bmatrix}\sqrt{\widetilde{\mathbf{D}}} & \mathbf{0}\end{bmatrix}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\begin{bmatrix}\sqrt{\widetilde{\mathbf{D}}} \\ \mathbf{0}\end{bmatrix}\mathbf{V}^\top)^2 \\ &= (\mathbf{I}_n - \mathbf{V}\sqrt{\widetilde{\mathbf{D}}}(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-1}\sqrt{\widetilde{\mathbf{D}}}\mathbf{V}^\top)^2 \\ &= (\mathbf{V}\mathbf{V}^\top - \mathbf{V}\widetilde{\mathbf{D}}(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-1}\mathbf{V}^\top)^2 \\ &= (\mathbf{V}(\mathbf{I}_n - \widetilde{\mathbf{D}}(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-1})\mathbf{V}^\top)^2 \\ &= \mathbf{V}\overline{\mathbf{\Lambda}}_n^2(\widetilde{\mathbf{D}} + \overline{\mathbf{\Lambda}}_n)^{-2}\mathbf{V}^\top,\end{aligned}$$

which yields the claim.

C.2 Proof of Theorem 2

For $\widehat{\theta} = \widehat{\theta}_{\mathbf{\Lambda}}(\mathbf{y})$, we claim that

$$\frac{1}{n}\mathbb{E}\left[\|\mathbf{X}\widehat{\theta} - \mathbf{X}\theta_\star\|^2\right] = \frac{1}{n}\sum_{i=1}^{\min\{n,d\}} \underbrace{\frac{\lambda_i^2\rho_i w_i^2 + \sigma^2\rho_i^2}{(\rho_i + \lambda_i)^2}}_{=:h_i(\lambda_i)} \tag{64}$$

Let us assume the claim as given at the moment, and prove equation (30a). Recalling the optimal choice $\lambda_{i,\mathrm{opt}}$ of the regularization parameter for MDL–COMP from equation (52), and noting that the in-sample MSE is separable in each term λ_i, it remains to show that

$$\operatorname*{argmin}_{\lambda_i \in [0,\infty)} h_i(\lambda_i) = \lambda_{i,\mathrm{opt}} = \frac{\sigma^2}{w_i^2} \tag{65}$$

To this end, we note that

$$h_i'(\lambda_i) = 0 \quad \Longleftrightarrow \quad 2\frac{\rho_i^2 w_i^2 \lambda_i}{(\rho_i + \lambda_i)^3} - 2\frac{\rho_i^2 \sigma^2}{(\rho_i + \lambda_i)^3} = 0 \quad \Longleftrightarrow \quad \widetilde{\lambda}_{i,\mathrm{opt}} = \frac{\sigma^2}{w_i^2}.$$

On the boundary of feasible values of $\lambda_i = [0, \infty)$, we have $h_i(0) = \sigma^2$, and $h_i(\lambda_i) \to \rho_i w_i^2$ as $\lambda_i \to \infty$. Noting that

$$h_i(\widetilde{\lambda}_i^{\mathrm{opt}}) = \frac{\rho_i w_i^2 \sigma^2}{\rho_i w_i^2 + \sigma^2} \leq \min\left\{\rho_i w_i^2, \sigma^2\right\},$$

yields the claim (65).

Next we establish the bound (30b) on the fixed design prediction error. Using λ_i as a convenient shorthand for $\lambda_{i,\mathrm{opt}}$, we substitute it into the RHS of equation (64), thereby finding that

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left[\|\mathbf{X}\widehat{\theta}_{\mathbf{\Lambda}_{\mathrm{opt}}} - \mathbf{X}\theta_\star\|^2\right] &= \frac{1}{n}\sum_{i=1}^{\min\{n,d\}} \frac{(\frac{\sigma^2}{w_i^2})^2 \rho_i w_i^2 + \sigma^2 \rho_i^2}{(\rho_i + (\frac{\sigma^2}{w_i^2}))^2} \\ &= \frac{\sigma^2}{n}\sum_{i=1}^{\min\{n,d\}} \frac{1}{1 + \frac{\sigma^2}{\rho_i w_i^2}}.\end{aligned}$$

We now make note of the elementary inequality $x \leq -\log(1-x)$ valid for $x \in [0,1)$. Applying this inequality with $x = (1 + \frac{\sigma^2}{\rho_i w_i^2})^{-1}$, we have $-\log(1-x) = \log(1 + \frac{\rho_i w_i^2}{\sigma^2})$, and hence

$$\frac{1}{n}\mathbb{E}\left[\|\mathbf{X}\widehat{\theta}_{\mathbf{\Lambda}_{\mathrm{opt}}} - \mathbf{X}\theta_\star\|^2\right] \leq \frac{\sigma^2}{n}\sum_{i=1}^{\min\{n,d\}} \log\left(1 + \frac{\rho_i w_i^2}{\sigma^2}\right) = \frac{\sigma^2}{n}\mathcal{R}_{\mathrm{opt}},$$

where we have used $\mathcal{R}_{\mathrm{opt}}$ as a shorthand for the optimal redundancy.

Proof of equation (64): There are two cases to be considered, namely $d > n$ and $d < n$. Both cases can be handled together by using the linear-algebraic claims (55) and (63) as needed. Following steps similar to those in the proof of Theorem 1, we find that

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{X}\widehat{\theta} - \mathbf{X}\theta_\star\|^2\right] &= \mathbb{E}\left[\|\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\mathbf{y} - \mathbf{X}\theta_\star\|^2\right] \\ &= \|\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\mathbf{X}\theta_\star - \mathbf{X}\theta_\star\|^2 + \mathbb{E}\left[\|\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \mathbf{\Lambda})^{-1}\mathbf{X}^\top\xi\|^2\right] \\ &\overset{(i)}{=} \|\mathbf{V}\sqrt{\mathbf{D}}((\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{D} - \mathbf{I})\mathbf{w}\|^2 + \mathbb{E}\left[\|\mathbf{V}\sqrt{\mathbf{D}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\sqrt{\mathbf{D}}\mathbf{V}^\top\xi\|^2\right] \\ &= \|\sqrt{\mathbf{D}}((\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\overline{\mathbf{\Lambda}})\mathbf{w}\|^2 + \sigma^2\,\mathrm{trace}\left(\sqrt{\mathbf{D}}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\mathbf{D}(\mathbf{D} + \overline{\mathbf{\Lambda}})^{-1}\sqrt{\mathbf{D}}\right) \\ &= \sum_{i=1}^{\min\{n,d\}}\left(\frac{\rho_i\lambda_i^2 w_i^2}{(\rho_i + \lambda_i)^2} + \frac{\sigma^2\rho_i^2}{(\rho_i + \lambda_i)^2}\right),\end{aligned}$$

and we are done.

C.3 Proof of Theorem 3

Let us introduce the shorthand $\mathbf{y}^\star = (f^\star(x_1), \ldots, f^\star(x_n))^\top$. Given data $\mathbf{y}$ generated according to the model (24), we have $\mathbf{y} \sim \mathcal{N}(\mathbf{y}^\star, \sigma^2 \mathbb{I}_n)$. Let $p(\mathbf{y}; \mathbf{y}^\star)$ denote the corresponding density of $\mathbf{y}$. With this notation, we can write

$$\begin{aligned}\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{f^\star} \,\|\, \mathbb{Q}_\lambda) &= \mathbb{E}_{\mathbf{y}}\left[\log \frac{p(\mathbf{y}; \mathbf{y}^\star)}{q_\lambda(\mathbf{y})}\right] \\ &= \underbrace{-\mathbb{E}_{\mathbf{y}}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{y}^\star\|^2\right]}_{=:T_1} + \underbrace{\mathbb{E}_{\mathbf{y}}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{K}\widehat{\theta}\|^2 + \frac{\lambda}{2\sigma^2}\widehat{\theta}^\top \mathbf{K}\widehat{\theta}\right]}_{=:T_2} + \underbrace{\log \mathrm{C}_\lambda}_{=:T_3}.\end{aligned}$$

Note that the term $T_1 = -n/2$ by definition of our noisy observation model. Turning to the term T_2, we substitute the expression (11) for $\widehat{\theta}$, thereby finding that

$$\begin{aligned}\|\mathbf{y} - \mathbf{K}\widehat{\theta}\|^2 + \lambda\|\widehat{\theta}\|^2 &= \mathbf{y}^\top \lambda^2 (\mathbf{K} + \lambda \mathbf{I})^{-2}\mathbf{y} + \lambda \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y} \\ &= \lambda \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}\end{aligned}$$

and thus

$$T_2 = \frac{1}{2\sigma^2}\mathbb{E}[\|\mathbf{y} - \mathbf{K}\widehat{\theta}\|^2 + \lambda\|\widehat{\theta}\|^2] = \frac{\lambda}{2\sigma^2}\mathbf{y}^{\star\top}(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}^\star + \frac{1}{2}\sum_{i=1}^n \frac{\lambda}{\rho_i + \lambda}.$$

Moreover, using an argument similar to that used in equation (61), we can write

$$T_3 = \log \mathrm{C}_\lambda = \log \sqrt{\det(\frac{1}{\lambda}(\mathbf{K} + \lambda \mathbf{I}))} = \frac{1}{2}\sum_{i=1}^n \log\left(\frac{\rho_i + \lambda_i}{\lambda_i}\right).$$

Using the eigenvalue decomposition $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ where $\mathbf{D} = \mathrm{diag}(\rho_1, \ldots, \rho_n)$, and setting $\alpha := \mathbf{U}^\top \mathbf{y}^\star$, we thus find that

$$\frac{1}{n}\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{f^\star} \,\|\, \mathbb{Q}_\lambda) = \frac{1}{2n}\sum_{i=1}^n \frac{\lambda}{\lambda + \rho_i}\left(\frac{\alpha_i^2}{\sigma^2} + 1\right) + \frac{1}{2n}\sum_{i=1}^n \log\left(\frac{\rho_i}{\lambda} + 1\right) - \frac{1}{2}. \tag{66}$$

The remaining proof make use of arguments similar to those in Raskutti et al. (2014, Lemma 7). Recall the decomposition (23) for the kernel $\mathcal{K}$. For any function $f^\star \in \mathbb{H}$, we can write

$$f^\star(x) = \sum_{k=1}^\infty \sqrt{\mu_k} a_k \phi_k(x) \quad \text{with} \quad a_k = \frac{1}{\sqrt{\mu_k}}\int f(z)\phi_k(z) d\nu(z) \quad \text{and} \quad \|f^\star\|_{\mathbb{H}}^2 = \sum_{k=1}^\infty a_k^2,$$

where ν denotes the marginal distribution of the covariates. Define the linear operators $\Psi_{\mathbf{X}} : \ell^2(\mathbb{N}) \to \mathbb{R}^n$ as $\Psi_{\mathbf{X}}[j, i] = \phi_i(x_j)$ for $i \in \mathbb{N}$ and $j \in [n]$ (the matrix $\Psi_{\mathbf{X}}$ has n rows and infinite columns). and the diagonal linear operator $\mathfrak{D} : \ell^2(\mathbb{N}) \to \ell^2(\mathbb{N})$ as $\mathfrak{D}[i, i] = \mu_i$ and $\mathfrak{D}[i, j] = 0$ if $i \neq j$ for $i, j \in \mathbb{N}$. Given this representation, we can rewrite the vector $\mathbf{y}^\star$ and the kernel matrix $\mathbf{K}$ as follows

$$\mathbf{y}^\star = \Psi_{\mathbf{X}}\mathfrak{D}^{\frac{1}{2}}a \quad \text{and} \quad \mathbf{K} = \Psi_{\mathbf{X}}\mathfrak{D}\Psi_{\mathbf{X}}^\top$$

Combining the latter representation with $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$, we find that there exists an operator $\Gamma : \ell^2(\mathbb{N}) \to \mathbb{R}^n$ with adjoint $\Gamma^* : \mathbb{R}^n \to \ell^2(\mathbb{N})$ and $\Gamma\Gamma^* = \mathbf{I}_n$ such that

$$\Psi_{\mathbf{X}}\mathfrak{D}^{\frac{1}{2}} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}\Gamma \implies \alpha = \mathbf{U}^\top \mathbf{y}^\star = \mathbf{U}^\top \Psi_{\mathbf{X}}\mathfrak{D}^{\frac{1}{2}} a = \mathbf{U}^\top \mathbf{U}\mathbf{D}^{\frac{1}{2}} \underbrace{\Gamma a}_{=:\beta} = \mathbf{D}^{\frac{1}{2}}\beta.$$

Note that $\beta \in \mathbb{R}^n$ and we have $\alpha_i^2 = \rho_i \beta_i^2$. Furthermore,

$$\sum_{i=1}^n \beta_i^2 = \|\Gamma a\|_2^2 \leq \|a\|_2^2 = \|f^\star\|_{\mathbb{H}}^2, \tag{67}$$

since Γ is a unitary operator. Thus, we can rewrite (66) as

$$\begin{aligned}
\frac{1}{n}\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{f^\star} \,\|\, \mathbb{Q}_\lambda) &= \frac{1}{2n}\sum_{i=1}^n \frac{\lambda}{\lambda+\rho_i}\left(\frac{\rho_i\beta_i^2}{\sigma^2}+1\right) + \frac{1}{2n}\sum_{i=1}^n \log\left(\frac{\rho_i}{\lambda}+1\right) - \frac{1}{2} \\
&= \frac{1}{2n}\lambda\sum_{i=1}^n \frac{\rho_i}{\lambda+\rho_i}\frac{\beta_i^2}{\sigma^2} + \frac{1}{2n}\sum_{i=1}^n \log\left(\frac{\rho_i}{\lambda}+1\right) + \frac{1}{2n}\sum_{i=1}^n \frac{\lambda}{\lambda+\rho_i} - \frac{1}{2} \\
&\overset{(i)}{\leq} \frac{\lambda}{2n}\left(\frac{\sum_{i=1}^n \beta_i^2}{\sigma^2}\right) + \frac{1}{2n}\sum_{i=1}^n \log\left(\frac{\rho_i}{\lambda}+1\right) \\
&\overset{(ii)}{\leq} \frac{\lambda}{2n}\frac{\|f^\star\|_{\mathbb{H}}^2}{\sigma^2} + \frac{1}{2n}\sum_{i=1}^n \log\left(\frac{\rho_i}{\lambda}+1\right),
\end{aligned} \tag{68}$$

where step (i) follows from the fact that $\max\{\rho_i, \lambda\}/(\lambda+\rho_i) \leq 1$, and step (ii) from the bound (67). The proof is now complete.

Remark: Let us discuss some scenarios in which the inequalities in (i) and (ii) are tight after optimizing the bound (68) over λ. Call the optimal choice λ_{opt}, and let $i^\star$ denote the index such that $\lambda_{\mathrm{opt}} \approx \rho_{i^\star}$. The inequality in step (i) is relatively tight when: (a) $\sum_{i=1}^n \lambda/(\lambda+\rho_i) \approx n$; and (b) $\|\beta\|_2^2 \approx \sum_{j \leq i^\star} \beta_j^2$. Condition (a) holds if the eigenvalue $\{\rho_i\}$ have suitablly rapid decay. whereas condition (b) holds when $i^\star < n$ and the $\{\beta_j\}$ sequence is suitably decaying.

In order for inequality (ii) to be relatively tight, we need the inequality (67) to be relatively tight. This property holds when we can obtain a good approximation of the sum (23) by summing only its first n terms; in this case, Γ and $\Gamma^\star$ are simply $n \times n$ unitary matrices. This property holds when either the eigenvalues $\{\mu_k\}$ decay quickly, or the kernel $\mathcal{K}$ has a finite rank strictly smaller than n, in which case $\mu_k = 0$ for $k \geq n$. Note that the different decay conditions covered in Corollary 1 are all sufficient for these kernel conditions to hold.

C.4 Proof of Theorem 4

Our result involves the function

$$g(a,b) := \frac{1}{4}\left(\sqrt{\mathrm{snr}(1+\sqrt{\gamma})^2+1} - \sqrt{\mathrm{snr}(1-\sqrt{\gamma})^2+1}\right)^2, \quad \text{for} \quad a, b > 0. \tag{69}$$

The proof of this theorem makes use of the analytical expression of MDL–COMP from Theorem 1 and few random matrix theory results. Let $\mathbb{F}_d : a \mapsto \frac{1}{d}\sum_{i=1}^{d} \mathbf{1}(a \leq \rho_i)$ denote the empirical distribution of the eigenvalues of the matrix $\mathbf{X}^\top\mathbf{X}$.

First, we note that since $\theta_\star$ is assumed to be drawn from rotationally invariant distribution, we can rewrite the MDL-complexity expression as

$$\begin{aligned}
\mathbb{E}_{\theta_\star}[\mathcal{R}_{\text{opt}}] = \mathbb{E}_{\mathbf{w}}\left[\frac{1}{n}\sum_{i=1}^{\min\{n,d\}} \log\left(1+\frac{\rho_i w_i^2}{\sigma^2}\right)\right] &\overset{(i)}{\leq} \left[\frac{1}{n}\sum_{i=1}^{\min\{n,d\}} \log\left(1+\frac{\rho_i \mathbb{E}_{\mathbf{w}}[w_i^2]}{\sigma^2}\right)\right] \\
&\overset{(ii)}{=} \frac{1}{n}\sum_{i=1}^{d} \log\left(1+\rho_i \cdot \text{snr}\right) \\
&= \frac{d}{n}\left(\frac{1}{d}\sum_{i=1}^{d} \log\left(1+\rho_i \cdot \text{snr}\right)\right) \\
&= \frac{d}{n}\cdot\int_{Z=0}^{\infty} \log(1+Z\cdot\text{snr}) d\mathbb{F}_d(Z),
\end{aligned}$$

where step (i) follows from Jensen's inequality, i.e., $\mathbb{E}[\log(1+W)] \leq \log(1+\mathbb{E}[W])$, and step (ii) follows from the fact that $\theta_\star$ is drawn from rotationally-invariant distribution and hence $\mathbf{w} = \mathbf{U}^\top\theta_\star \overset{d}{=} \theta_\star$, and that $\mathbb{E}[w_i^2] = \mathbb{E}[\|\theta_\star\|^2]/d$.

Next, we recall a standard result from random matrix theory. Under the assumptions of Theorem 4, as $d, n \to \infty$ with $d/n \to \gamma$, the empirical distribution $\mathbb{F}_d$ of the eigenvalues of the matrix $\mathbf{X}^\top\mathbf{X}$ converges weakly, almost surely to a nonrandom limiting distribution $\mathbb{MP}_\gamma$ with density

$$m_\gamma(a) = \left(1-\frac{1}{\gamma}\right)_+ \delta(a) + \frac{\sqrt{(a-b_1)_+(b_2-a)_+}}{2\pi\gamma a}, \quad \text{where } b_1 = (1-\sqrt{\gamma})^2 \text{ and } b_2 = (1+\sqrt{\gamma})^2. \tag{70}$$

This distribution is also known as the Marčenko-Pastur law with ratio index γ. (For background, see the papers (Marcenko and Pastur, 1967; Silverstein, 1995), or Theorem 2.35 in the book (Tulino and Verdú, 2004))

Next, we claim that as $d, n \to \infty$ with $d/n \to \gamma$, we have

$$\mathcal{R}_{\text{opt}} = \frac{d}{n}\cdot\int_{0}^{\infty} \log(1+Z\cdot\text{snr}) d\mathbb{F}_d(Z) \longrightarrow \gamma\cdot\mathbb{E}_{Z\sim\mathbb{MP}_\gamma}\left[\log(1+\text{snr}\cdot Z)\right], \tag{71}$$

almost surely. Assuming this claim as given at the moment, let us now complete the proof. In general, the analytical expressions for expectations under the distribution $\mathbb{MP}_\gamma$ are difficult to compute. However, the expectation $\mathbb{E}_{Z\sim\mathbb{MP}_\gamma}[\log(1+\gamma Z)]$ is known as the Shannon transform and there exists a closed form for it (see Example 2.14 (Tulino and Verdú, 2004))[18]:

$$\mathbb{E}_{Z\sim\mathbb{MP}_\gamma}\left[\log(1+\text{snr}\cdot Z)\right] =: \mathcal{V}_Z(\text{snr}) = \log\left(1+\text{snr}-\delta\right) + \frac{1}{\gamma}\log\left(1+\gamma\cdot\text{snr}-\delta\right) - \frac{\delta}{\text{snr}\cdot\gamma}, \tag{72}$$

[18] The transform in the reference (Tulino and Verdú, 2004) uses the $\log_2$ (base-2 logarithm) notation. The results stated here have been adapted for $\log_e$ (base-e, natural logarithm) in a straightforward manner.

where $\delta = g(\text{snr}, \gamma)$ and the function g was defined in equation (69). Putting the equations (71) and (72) together yields the expression (43) stated in Theorem 4.

Proof of claim (71): This part makes use of the Portmanteau theorem and some known results from Bai and Yin (2008). Recalling the constant $b_2 := (1+\sqrt{\gamma})^2$ from equation (70), for $a \in [0, \infty)$, consider the function $f(a)\log(1+\text{snr}\cdot a)\mathbf{1}(a \leq 2b_2)$ As $n, d \to \infty$ with $d/n \to \gamma$, we have

$$\int_0^\infty f(Z) d\mathbb{F}_d(Z) \overset{(i)}{=} \int_0^{2b_2} f(Z) d\mathbb{F}_d(Z) \overset{(ii)}{\longrightarrow} \int_0^{2b_2} f(Z) d\mathbb{MP}_\gamma(Z) \overset{(iii)}{=} \int_0^\infty f(Z) d\mathbb{MP}_\gamma(Z)$$
$$= \mathbb{E}_{Z\sim\mathbb{MP}_\gamma}[1+\log(\text{snr}\cdot Z)],$$

where steps (i) and (iii) follow from the definition of f, i.e., $f(Z) = 0$ for any $Z > 2b+2$. Moreover, step (ii) follows from applying the Portmanteau theorem, which states that the weak almost sure convergence (70) implies the convergence in expectation for any bounded function, and f is a bounded function. Finally, we argue that for large enough n, we have

$$\int_0^\infty \log(1+Z\cdot\text{snr}) d\mathbb{F}_d(Z) \overset{(iv)}{=} \int_0^{2b_2} \log(1+Z\cdot\text{snr}) d\mathbb{F}_d(Z) \overset{(v)}{=} \int_0^\infty f(Z) d\mathbb{F}_d(Z)$$

In order to establish step (iv), we invoke a result from the work (Bai and Yin, 2008) (Corollary 1.8). It states that for large enough n, the largest eigenvalue of the matrix $\mathbf{X}^\top\mathbf{X}$ is bounded above by $2(1+\sqrt{\gamma})^2$ almost surely, and thus, we can restrict the limits of the integral on the LHS to the interval $[0, 2b_2]$. Step (v) follows trivially from the definition of the function f. Putting the pieces together yields the claim (71).

C.5 Proof of Theorem 5

The proof of this theorem makes use of the algebra used in previous proofs, and hence we simply illustrate the main steps. We have

$$\mathbb{E}\left[\log\left(\frac{1}{q_{\mathbf{\Lambda}}(\mathbf{y})}\right)\right] = \mathbb{E}\left[\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta}\right] + \log \mathrm{C}_{\mathbf{\Lambda}},$$

where $\log \mathrm{C}_{\mathbf{\Lambda}} = \frac{1}{2}\sum_{i=1}^{\min\{n,d\}} \log\left(\frac{\rho_i+\lambda_i}{\lambda_i}\right)$. Repeating arguments similar to those in equations (57) and (58), we find that

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{X}\widehat{\theta}\|^2 + \frac{1}{2\sigma^2}\widehat{\theta}^\top\mathbf{\Lambda}\widehat{\theta}\right] \\
&= \frac{1}{2\sigma^2}\theta_\star^\top\mathbf{X}^\top\left(\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right)\mathbf{X}\theta_\star + \frac{1}{2\sigma^2}\,\text{trace}\left[\text{Var}(\mathbf{y}|\mathbf{X})\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right] \\
&\qquad + \frac{1}{2\sigma^2}\,\text{trace}\left[\text{Var}(\mathbf{y}|\mathbf{X})(\mathbf{I}-\mathbf{V}\mathbf{V}^\top)\right] \\
&\overset{(i)}{\leq} \frac{1}{2\sigma^2}\left(\mathbf{w}^\top\mathbf{D}\mathbf{\Lambda}(\mathbf{D}+\mathbf{\Lambda})^{-1}\mathbf{w}\right) + \frac{1}{2\sigma^2}\,\text{trace}\left[\sigma^2\mathbf{I}_n\text{Var}(\mathbf{y}|\mathbf{X})\mathbf{V}\overline{\mathbf{\Lambda}}(\mathbf{D}+\overline{\mathbf{\Lambda}})^{-1}\mathbf{V}^\top\right] \\
&\qquad + \frac{1}{2\sigma^2}\,\text{trace}\left[\sigma^2\mathbf{I}_n(\mathbf{I}-\mathbf{V}\mathbf{V}^\top)\right] \\
&= \frac{(n-\min\{n,d\})}{2} + \frac{1}{2}\sum_{i=1}^d\left(\frac{\rho_i w_i^2}{\sigma^2}+1\right)\frac{\lambda_i}{\lambda_i+\rho_i}
\end{aligned}$$

where step (i) follows from the following linear algbera fact: For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \succeq 0$, if $\mathbf{A} \succeq \mathbf{B}$, then trace$[\mathbf{AC}] \geq$ trace$[\mathbf{BC}]$. Moreover, noting that in step (i), we have equality when $\mathrm{Var}(\mathbf{y}|\mathbf{X}) = \sigma^2 \mathbf{I}_n$. Putting the pieces together, we conclude that

$$\max_{\mathbb{P} \in \mathcal{P}_{\text{bndvar}}} \mathbb{E}\left[\log\left(\frac{1}{q_{\mathbf{\Lambda}}(\mathbf{y})}\right)\right] = \frac{(n - \min\{n,d\})}{2} + \frac{1}{2}\sum_{i=1}^{d}\left(\frac{\rho_i w_i^2}{\sigma^2} + 1\right)\frac{\lambda_i}{\lambda_i + \rho_i} + \frac{1}{2}\sum_{i=1}^{\min\{n,d\}} \log\left(\frac{\rho_i + \lambda_i}{\lambda_i}\right).$$

As function of $\{\lambda_i\}$ in comparison to the function equation (51), this objective is a simply shifted by a constant. Consequently, it admits the same minimizer

$$\arg \min_{\mathbb{Q}_{\mathbf{\Lambda}} \in \mathcal{Q}_{\text{Ridge}}} \max_{\mathbb{P} \in \mathcal{P}_{\text{bndvar}}} \mathbb{E}\left[\log\left(\frac{1}{q_{\mathbf{\Lambda}}(\mathbf{y})}\right)\right] = \mathbb{Q}_{\mathbf{\Lambda}_{\text{opt}}},$$

as claimed.

C.6 Proof of Corollary 1

We prove the bounds for polynomial and exponential decay of eigenvalues separately.

C.6.1 Proof with polynomial decay of eigenvalues

When $\mathcal{K}(x,x) = 1$, we have trace$(\mathbf{K}) = \sum_{i=1}^n \rho_i = n$. Thus, with $\rho_i \precsim i^{-2\alpha}$, we can conclude that $\rho_i \leq c_\alpha n i^{-2\alpha}$ where $c_\alpha := \sum_{j=1}^n j^{-2\alpha} \leq 1/(2\alpha - 1)$, where the additional factor of n arises due to the constraint that trace$(\mathbf{K}) = \sum_{i=1}^n \rho_i = n$.

Since the MDL-COMP is an increasing function in ρ_i, it suffices to consider the case when $\rho_i = n c_\alpha i^{-2\alpha}$. Setting $\alpha = \omega/d$ covers the case of $\rho_i \precsim i^{-2\omega/d}$, and setting $\alpha = \frac{1}{2}(d+a)$ covers the case of $\rho_i \precsim i^{-(d+a)}$. With the assumptions $\omega > d/2$ and $d + a > 1$, we find that for both cases $2\alpha - 1 > 0$ and hence c_α is finite.

Let us write λ as $nM^{-2\alpha}$, and given the rapid decay of ρ_i, we can use the bounds

$$\sum_{i=1}^{\lceil M \rceil} \log(\rho_i/\lambda + 1) \leq \sum_{i=1}^{\lceil M \rceil} \log((M/i)^{2\alpha} + 1) \overset{(i)}{\leq} c_1 \cdot \alpha M \log M$$

$$\sum_{i=\lceil M \rceil + 1}^{n} \log(\rho_i/\lambda + 1) \leq \sum_{i=\lceil M \rceil + 1}^{n} \log((M/i)^{2\alpha} + 1) \leq \sum_{i=\lceil M \rceil + 1}^{n} (M/i)^{2\alpha} \leq c_\alpha.$$

where step (i) uses the fact that $\int_1^x \log z dz = x \log x - x \leq x \log x$ for $x > 1$, and step (ii) uses the fact that $\sum_{i=\lceil M \rceil + 1}^n (M/i)^{2\alpha} \leq \sum_{j=1}^\infty j^{-2\alpha} = c_\alpha$. Thus, we can write the RHS in equation (31) as

$$\frac{\lambda}{2n}\frac{\|f^\star\|_{\mathbb{H}}^2}{\sigma^2} + \frac{1}{2n}\sum_{i=1}^n \log\left(\frac{\rho_i}{\lambda} + 1\right) \leq \frac{1}{2n}\left[nM^{-2\alpha}\mathrm{SNR}^2 + c\alpha M \log M + c_\alpha\right]$$

In order to minimize the RHS above with respect to M, we can equate the two terms inside the brackets, and obtain

$$\frac{n\mathrm{SNR}^2}{\alpha} \asymp M^{1+2\alpha}\log M \quad \Longleftrightarrow \quad M \asymp \left(\frac{n\mathrm{SNR}^2/\alpha}{\log(n\mathrm{SNR}^2/\alpha)}\right)^{1/(1+2\alpha)}$$

which yields

$$\begin{aligned}\text{MDL–COMP} \asymp M^{-2\alpha}\text{SNR}^2 &= \left(\frac{n\text{SNR}^2/\alpha}{\log(n\text{SNR}^2/\alpha)}\right)^{-2\alpha/(1+2\alpha)} \text{SNR}^2 \\ &= c'_\alpha \left(\frac{\log(n\text{SNR}^2)}{n}\right)^{2\alpha/(1+2\alpha)} \text{SNR}^{2/(1+2\alpha)} \\ &= C_{\alpha,\text{SNR}} \left(\frac{\log(n\text{SNR}^2)}{n}\right)^{2\alpha/(1+2\alpha)},\end{aligned}$$

where

$$C_{\alpha,\text{SNR}} = \alpha^{2\alpha/(1+2\alpha)}\text{SNR}^{2/(1+2\alpha)}. \tag{73}$$

Setting $\alpha = \omega/d$ and $\frac{1}{2}(d+a)$ respectively recovers the first two bounds stated in the display (32).

C.6.2 Proof with exponential decay of eigenvalues

Now, we do a reparametrization of λ as $n\exp(-M/d)$, and given the rapid decay of ρ_i, we use the following bounds:

$$\begin{aligned}\sum_{i=1}^{\lceil M\rceil} \log(\rho_i/\lambda + 1) \le \sum_{i=1}^{\lceil M\rceil} \log(e^{(M-i)/d}+1) &\overset{(i)}{\le} \frac{c}{d}\cdot\sum_{i=1}^{\lceil M\rceil}(M-i) \le c'\frac{M^2}{d} \\ \sum_{i=\lceil M\rceil+1}^{n} \log(\rho_i/\lambda+1) = \sum_{i=\lceil M\rceil+1}^{n} \log(e^{(M-i)/d}+1) &\le \sum_{i=\lceil M\rceil+1}^{n} e^{(M-i)/d} \le c''_d.\end{aligned}$$

where we use the fact that $\log(1+e^x) \le cx$ for large x, and $\log(1+e^x) \le e^x$ for all x. Thus, we can write the RHS in equation (31) as

$$\frac{\lambda}{2n}\frac{\|f^\star\|_{\mathbb{H}}^2}{\sigma^2} + \frac{1}{2n}\sum_{i=1}^{n}\log\left(\frac{\rho_i}{\lambda}+1\right) \le \frac{1}{2n}\left[ne^{-M/d}\text{SNR}^2 + c'\frac{M^2}{d} + c''_d\right]$$

In order to minimize the RHS of the above display with respect to M, we can equate the two terms inside the brackets, and obtain

$$dn\text{SNR}^2 \asymp e^{M/d}M^2 \quad\Longleftrightarrow\quad M \asymp d\log\left(\frac{nd\text{SNR}^2}{d\log(nd\text{SNR}^2)}\right).$$

Thus, we have

$$\text{MDL–COMP} \asymp \frac{M^2}{dn} = \frac{d\log^2(nd\text{SNR}^2)}{n},$$

as claimed.

Appendix D. Bias-variance tradeoff: Role of estimator and design matrix

In this appendix, we show that the bias-variance tradeoff for OLS heavily depends on the design matrix. More precisely, depending on the structure of the design matrix, it is possible to observe double-descent or multiple descent, where the peaks can occur at values of d/n not necessarily equal to 1 in the test MSE when varying d. While OLS can exhibit a wide range of behavior, we also show that the test MSE for CV-tuned ridge regression exhibits the familiar U-shaped behavior for all the choices of design considered here.

In the experimental results shown here, we generate data from a linear Gaussian model of the form

$$\mathbf{y} = \underbrace{\mathbf{X}\theta_\star}_{\mathbf{y}_\star} + \xi, \tag{74}$$

where $\xi \in \mathbb{R}^n \sim \mathcal{N}(0, \sigma \mathbf{I}_n)$ with $\sigma = 0.1$. In all cases, we keep the training sample size fixed at $n = 200$, and the maximum size of covariates is fixed at 2000. With these fixed choices, we consider several different choices of the design matrix $\mathbf{X}$, along with two choices for the unknown regression vector $\theta_\star$.

Choice for X: We consider two possible random ensembles for the design matrix $\mathbf{X}$. Let $\mathbf{A} \in \mathbb{R}^{200 \times 2000}$ denote a matrix whose entries are drawn iid from $\mathcal{N}(0, 1)$. Let $\mathbf{B} \in \mathbb{R}^{2000 \times 2000}$ denote a diagonal matrix such that $\mathbf{B}_{ii} = |\cos i|$ for i even, and 0 otherwise. Then the two choices for the design matrix $\mathbf{X}$ are (I) *Gaussian design* with $\mathbf{X} = \mathbf{A}$, and (II) *Cosine design* with $\mathbf{X} = \mathbf{AB}$.

Choice for $\theta_\star$: In parallel, we consider two possible random ensembles for the unknown regression vector $\theta_\star$. In all cases, the response $\mathbf{y}_\star$ depends on only on $d_\star$ covariates in total. In setting (A) where $\mathbf{d}_\star < \mathbf{n}$, we choose $\theta_\star$ to have non-zero entries for the index $\{11, 12, \ldots, 60\}$, i.e., the true dimensionality of the dataset is $d_\star = 60$, which is less than the sample size $n = 200$. In setting (B) where $\mathbf{d}_\star > \mathbf{n}$, we choose $\theta_\star$ to have non-zero entries for all indices in the set $\{1, 2, \ldots, 400\}$. Consequently, the number of free parameters $d_\star = 400$ is much larger than the sample size $n = 200$. In both settings, the non-zero entries of $\theta_\star$ are drawn iid from $\mathcal{N}(0, 1)$, and then normalized such that $\|\theta_\star\| = 1$.

Taking all possible combinations of random ensembles for $\mathbf{X}$ and $\theta_\star$ yields four distinct experimental settings: IA, IB, IIA, IIB. Given a particular setting—for instance, setting IA with Gaussian design and $d_\star = 60 < n$ (IA)—we generate a dataset of n observations and then fit different estimators with a varying number of covariates (denoted by d); i.e., the response variable $\mathbf{y}$ remains fixed and we only vary the dimensionality of the design matrix for fitting OLS or Ridge model. We then compute the test MSE (computed on an independent draw of $\mathbf{y}_\star^{\text{test}}$ of size $n_{\text{test}} = 1000$). We redraw the noise in the observation 50 times ($\mathbf{y}_\star^{\text{train}}$ and $\mathbf{y}_\star^{\text{test}}$ remain fixed), and plot the average of test MSE over these runs. For a given d, the bias and variance for a given estimator (OLS or Ridge) are computed as

follows:

$$\text{Bias}(d) = \frac{1}{n_{\text{test}}} \|\mathbf{y}_\star^{\text{test}} - \overline{\mathbf{y}^{\text{est}}}\|^2, \quad \text{and}$$
$$\text{Variance}(d) = \frac{1}{50} \sum_{r=1}^{50} \frac{1}{n_{\text{test}}} \|\mathbf{y}_r^{\text{est}} - \overline{\mathbf{y}^{\text{est}}}\|^2,$$

where r denotes the index of the experiment (redraw of noise ξ), $\mathbf{y}_r^{\text{est}}$ denotes the estimate for the response for the test dataset for the r-th experiment, and $\overline{\mathbf{y}^{\text{est}}}$ denotes the average of the predictions made by the estimator across all 50 experiments.

The four panels in Fig. F1 show, for each of these four settings, plots of the radio d/n versus test MSE for the OLS and CV-tuned ridge estimators. Fig. F2 shows the underlying bias-variance tradeoff curves in settings IA and IIA.

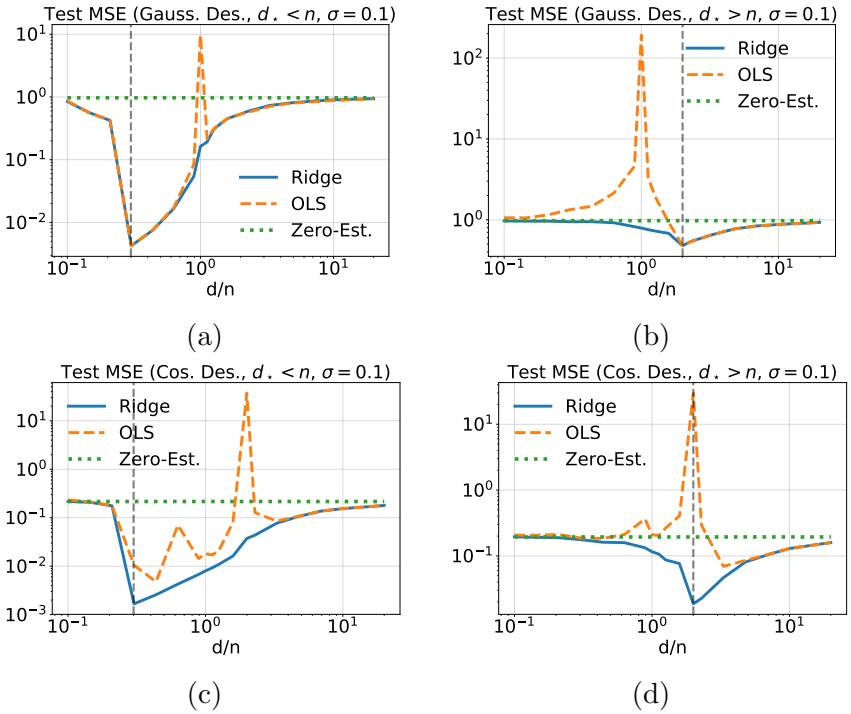

Figure F1. Plots of test MSE for OLS and CV-tuned Ridge for the four designs discussed in Appendix D, versus the dimensionality of the covariates in the fitted model varies. The training sample size is fixed at $n = 200$.

From these figures, we can draw the following conclusions. On one hand, the test MSE of the CV-tuned ridge estimator exhibits the classical U-shaped curve for all the settings. On the other hand, the behavior of OLS is heavily dependent on the covariate design along with the choice of $d_\star$. For $d < d_\star$, both OLS and Ridge show the classical tradeoff between bias (deccreasing in d) and variance (increasing in d). For $d > d_\star$, the bias of the tuned ridge increases monotonically with d, but the variance increases until a design-dependent threshold, after which it then decreases. In almost all cases, the bias of OLS monotonically increases

as d increases above $d_\star$; however, the variance term can show multiple peaks depending on the design, and these peaks frequently show up in the test MSE as well.

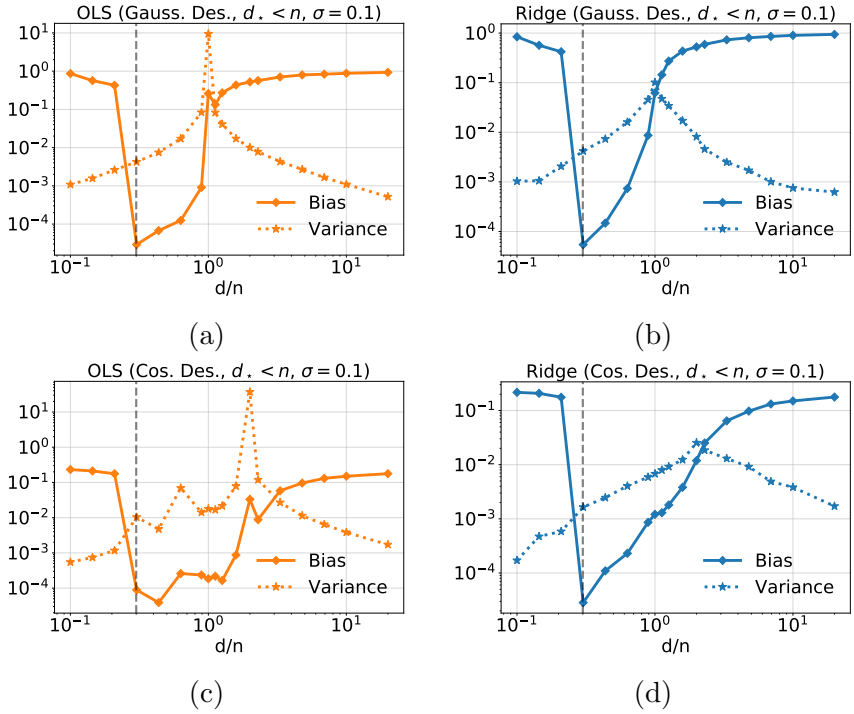

Figure F2. Bias-variance tradeoff of OLS and CV-tuned Ridge estimators for various designs discussed in Appendix D. Panels (a) and (b) show the results for design IA corresponding to the Fig. F1(a), and panels (c) and (d) show it for design IIA corresponding to the Fig. F1(c).

References

Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Hirotugu Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7411–7422, 2019.

Zhi-Dong Bai and Yong-Qua Yin. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. In *Advances In Statistics*, pages 108–127. World Scientific, 2008.

Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

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

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. *arXiv preprint arXiv:1802.01396*, 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019a.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019b.

Christoph Bernau, Markus Riester, Anne-Laure Boulesteix, Giovanni Parmigiani, Curtis Huttenhower, Levi Waldron, and Lorenzo Trippa. Cross-study validation for the assessment of prediction algorithms. *Bioinformatics*, 30(12):i105–i112, 2014.

Alberto Bietti and Francis Bach. Deep equals shallow for ReLU networks in kernel regimes. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=aDjoksTpXOP.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

Léonard Blier and Yann Ollivier. The description length of deep learning models. In *Advances in Neural Information Processing Systems*, pages 2216–2226, 2018.

Peter Bühlmann and Bin Yu. Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510, 1989.

Lee H Dicker. Ridge regression and asymptotic minimax estimation over spheres of growing dimension. *Bernoulli*, 22(1):1–37, 2016.

Edgar Dobriban and Stefan Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279, 2018.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Bradley Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.

Bradley Efron. The estimation of prediction error: covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–632, 2004.

Noureddine El Karoui. On information plus noise kernel random matrices. *The Annals of Statistics*, 38(5):3191–3216, 2010a.

Noureddine El Karoui. The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1): 1–50, 2010b.

Dean P Foster and Robert A Stine. The contribution of parameters to stochastic complexity. *Advances in Minimum Description Length Theory and Applications*, pages 195–213, 2004.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

Peter Grünwald and Teemu Roos. Minimum description length revisited. *International journal of mathematics for industry*, 11(01):1930001, 2019.

Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.

Peter D Grünwald and Nishant A Mehta. A tight excess risk bound via a unified PAC-Bayesian-Rademacher-Shtarkov-MDL complexity. *arXiv preprint arXiv:1710.07732*, 2017.

Peter D Grünwald and Nishant A Mehta. A tight excess risk bound via a unified PAC-Bayesian–Rademacher–Shtarkov–MDL complexity. In *Algorithmic Learning Theory*, pages 433–465. PMLR, 2019.

Mark H Hansen and Bin Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Trevor J Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019.

Geoffrey Hinton and Drew Van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th Annual ACM Conference on Computational Learning Theory*. Citeseer, 1993.

Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pages 9–1. JMLR Workshop and Conference Proceedings, 2012.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.

Lucas Janson, William Fithian, and Trevor J Hastie. Effective degrees of freedom: a flawed metaphor. *Biometrika*, 102(2):479–485, 2015.

S Kaufman and S Rosset. When does more regularization imply fewer degrees of freedom? sufficient conditions and counterexamples. *Biometrika*, 101(4):771–784, 2014.

Andrei N Kolmogorov. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 369–376, 1963.

Andrei Nikolaevich Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2(1-4):157–168, 1968.

Thomas CM Lee. An intrductionto coding theory iafid the two-part minimumljescription length'. *International Statistical Review*, 69(2):169–133, 2001.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018a.

Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

Xingguo Li, Junwei Lu, Zhaoran Wang, Jarvis Haupt, and Tuo Zhao. On tighter generalization bound for deep neural networks: CNNs, resnets, and beyond. *arXiv preprint arXiv:1806.05159*, 2018b.

X Liu, S Zheng, and X Feng. Estimation of error variance via ridge regression. *Biometrika*, 107(2): 481–488, 2020.

Marco Loog, Tom Viering, Alexander Mey, Jesse H Krijthe, and David MJ Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.

David JC MacKay. Bayesian nonlinear modeling for the prediction competition. *ASHRAE Transactions*, 100(2):1053–1062, 1994.

Colin L Mallows. Some comments on C_p. *Technometrics*, 15(4):661–675, 1973.

Vladimir A Marcenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

J Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of The Royal Society*, pages 4–415, 1909.

Mary Meyer and Michael Woodroofe. On the degrees of freedom in shape-restricted regression. *The Annals of Statistics*, 28(4):1083–1104, 2000.

Kohei Miyaguchi and Kenji Yamanishi. High-dimensional penalty selection via minimum description length principle. *Machine Learning*, 107(8-10):1283–1302, 2018.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 2020.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Preetum Nakkiran, Prayaag Venkat, Sham Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. *arXiv preprint arXiv:2003.01897*, 2020.

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

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. *International Conference on Learning Representations*, page To appear, 2019.

Shinji Nishimoto, An T Vu, Thomas Naselaris, Yuval Benjamini, Bin Yu, and Jack L Gallant. Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19):1641–1646, 2011.

Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

Randal S Olson, William La Cava, Patryk Orzechowski, Ryan J Urbanowicz, and Jason H Moore. PMLB: A large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10(1):36, 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *The Journal of Machine Learning Research*, 15(1): 335–366, 2014.

Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, 11(2):416–431, 1983.

Jorma Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, pages 1080–1100, 1986.

Gabriele Santin and Robert Schaback. Approximation of eigenfunctions in kernel-based spaces. *Advances in Computational Mathematics*, 42(4):973–993, 2016.

Jürgen Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

Xiaotong Shen and Jianming Ye. Adaptive model selection. *Journal of the American Statistical Association*, 97(457):210–221, 2002.

Yurii Mikhailovich Shtar'kov. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23(3):3–17, 1987.

Jack W Silverstein. Strong convergence of the empirical distribution of eigenvalues of large dimensional random matrices. *Journal of Multivariate Analysis*, 55(2):331–339, 1995.

Jeffrey S Simonoff. *Analyzing categorical data*. Springer Science & Business Media, 2013.

Marina Skurichina and Robert PW Duin. Bagging for linear classifiers. *Pattern Recognition*, 31(7): 909–930, 1998.

Marina Skurichina and Robert PW Duin. Bagging and the random subspace method for redundant feature spaces. In *International Workshop on Multiple Classifier Systems*, pages 1–10. Springer, 2001.

Marina Skurichina and Robert PW Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002.

Charles J Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, pages 1040–1053, 1982.

Yoshiki Tanaka, Kazuhisa Iwamoto, and Kuniaki Uehara. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*, 58(2-3):269–300, 2005.

Ryan J Tibshirani. Degrees of freedom and model search. *Statistica Sinica*, pages 1265–1296, 2015.

Ryan J Tibshirani and Jonathan Taylor. Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232, 2012.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *arXiv preprint arXiv:2009.14286*, 2020.

Antonia M Tulino and Sergio Verdú. Random matrix theory and wireless communications. *Foundations and Trends® in Communications and Information Theory*, 1(1):1–182, 2004.

Sara Van De Geer. *Empirical Processes in M-estimation*. Cambridge University Press, 2006.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.2641198.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer, 2015.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, and SciPy 1. 0 others. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 2020. doi: https://doi.org/10.1038/s41592-019-0686-2.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Bin Yu and Karl Kumbier. Veridical data science. *Proceedings of the National Academy of Sciences*, 117(8):3920–3929, 2020.

Bo Zhang, Xiaotong Shen, and Sunni L Mumford. Generalized degrees of freedom and adaptive model selection in linear mixed-effects models. *Computational Statistics & Data Analysis*, 56(3): 574–586, 2012.

Tong Zhang. From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 34(5):2180–2210, 2006.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16 (1):3299–3340, 2015.

Hui Zou, Trevor Hastie, and Robert Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.