Understanding Pathologies of Deep Heteroskedastic Regression

Eliot Wong-Toi
Department of Statistics
University of California, Irvine
ewongtoi@uci.edu

Alex Boyd
Department of Statistics
University of California Irvine
alexjb@uci.edu

Vincent Fortuin
Helmholtz AI, Munich
vincent.fortuin@helmholtz-munich.de

Stephan Mandt
Departments of Computer Science & Statistics
University of California, Irvine
mandt@uci.edu

Abstract

Recent studies have reported negative results when using heteroskedastic neural regression models. In particular, for overparameterized models, the mean and variance networks are powerful enough to either fit every single data point, or to learn a constant prediction with an output variance exactly matching every predicted residual, explaining the targets as pure noise. We study these difficulties from the perspective of statistical physics and show that the observed instabilities are not specific to any neural network architecture but are already present in a *field theory* of an overparameterized conditional Gaussian likelihood model. Under light assumptions, we derive a *nonparametric free energy* that can be solved numerically. The resulting solutions show excellent qualitative agreement with empirical model fits on real-world data and, in particular, prove the existence of *phase transitions*, i.e., abrupt, qualitative differences in the behaviors of the regressors upon varying the regularization strengths on the two networks. Our work provides a theoretical explanation for the necessity to carefully regularize heteroskedastic regression models. Moreover, the insights from our theory suggest a scheme for optimizing this regularization which is quadratically more efficient than the naïve approach.

1 Introduction

Regression and classification problems lie at the heart of deep learning (1; 2; 3). Heteroskedastic regression models assume that the output noise may depend on the input features x, and try to learn a conditional distribution $p(y|x)$ with non-uniform variance. This allows the model to assign different importances to training data and ultimately results in a model that "knows where it fails" (4; 5). Learning overparameterized heteroskedastic regression models (e.g. with deep neural networks) has proven to be difficult due to extreme forms of overfitting (6; 7) in the mean model, standard deviation model, or both as shown in Fig. 1. While several practical solutions to learning overparameterized heteroskedastic regression models have been proposed (4; 8; 9; 10), no comprehensive theoretical study of the failure of these methods has been offered so far.

We provide a theoretical analysis of the failure of heteroskedastic regression models in the overparameterized limit. To this end, we borrow a tool that abstracts away from any details of the involved neural network architectures: classical field theory from statistical mechanics (11; 12). Via our field-theoretical description, we can recover the optimized heteroskedastic regressors as solutions to

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

partial differential equations which can be solved numerically by optimizing the field theory's free energy functional.

2 A Field Theory for Overparameterized Heteroskedastic Regression

Heteroskedastic Regression Suppose we have a set of independent data points $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^N$ with covariates $x_i \in \mathcal{X} \subset \mathbb{R}^d$ drawn from a distribution $x_i \sim p(x)$ and responses $y_i \in \mathcal{Y} \equiv \mathbb{R}$ where $y_i \sim \mathcal{N}(\mu_i, \Lambda_i^{-\frac{1}{2}})$. We assume to be in a *heteroskedastic* setting, in which Λ_i need not equal Λ_j for $i \neq j$. Finally, we assume *both* the mean and standard deviation of y_i to be explainable via x_i: $y_i \,|\, x_i \sim \mathcal{N}(\mu(x_i), \Lambda(x_i)^{-\frac{1}{2}})$ for $i = 1, \ldots, N$ with continuous functions $\mu : \mathcal{X} \to \mathbb{R}$ and $\Lambda : \mathcal{X} \to \mathbb{R}_{>0}$.

Neural networks are well-known *universal function approximators*, which make them a reasonable choice for estimating μ, Λ (13). Let the mean network $\hat{\mu}_\theta : \mathcal{X} \to \mathbb{R}$ and precision (inverse-variance) network $\hat{\Lambda}_\phi : \mathcal{X} \to \mathbb{R}_{>0}$ be arbitrary depth, overparameterized feed-forward neural networks parameterized by θ and ϕ respectively. For a given input x_i, these networks collectively represent a corresponding predictive distribution for y_i: $\hat{p}(y_i \,|\, x_i) := \mathcal{N}(y_i; \hat{\mu}_\theta(x_i), \hat{\Lambda}_\phi(x_i)^{-\frac{1}{2}})$.

Pitfalls of MLE The maximum likelihood objective (MLE) in our setting is:

$$\mathcal{L}(\theta, \phi) \approx \frac{1}{2N} \sum_{i=1}^{N} \hat{\Lambda}_\phi(x_i)(y_i - \hat{\mu}_\theta(x_i))^2 - \log \hat{\Lambda}_\phi(x_i). \tag{1}$$

Given an infinitely flexible model, this objective function is ill-posed. The mean function $\hat{\mu}_\theta$ has no regularization, so during training it can estimate y to arbitrary precision for at least one data point (x_i, y_i). Then the residual corresponding, $y_i - \hat{\mu}_\theta(x_i) \to 0$ and the regularization for $\hat{\Lambda}_\phi$ vanishes, at least at the point x_i. If training reaches this point, the objective function becomes completely unstable due to effectively containing a term whose limit naïvely yields $\infty - \infty$.

We posit that additional regularization on $\hat{\Lambda}_\phi$, is required to avoid this instability. It can be tempting to think that one must regularize θ in order to avoid overfitting. And while this is generally true, the objective function $\mathcal{L}$ will still be unstable so long as *at least* one input x_i yields a perfect prediction. To prevent this from happening, we can include L_2 penalty terms for both θ and ϕ in our loss function:

$$\mathcal{L}_{\alpha,\beta}(\theta, \phi) := \mathcal{L}(\theta, \phi) + \alpha||\theta||_2^2 + \beta||\phi||_2^2 \tag{2}$$

where $\alpha, \beta > 0$ are penalty coefficients. As $\alpha \to \infty$, the network models a constant mean and as $\beta \to \infty$ we effectively model a homoskedastic regime. For computational reasons, we propose the following one-to-one reparameterization of the regularization coefficients as:

$$\mathcal{L}_{\rho,\gamma}(\theta, \phi) := \rho\mathcal{L}(\theta, \phi) + (1 - \rho)\left[\gamma||\theta||_2^2 + (1 - \gamma)||\phi||_2^2\right] \tag{3}$$

where $\rho, \gamma \in (0, 1)$. Since ρ, γ are bounded we are able to cover the space of regularization combinations by searching over $(0, 1)$ whereas in the α, β parameterization $\alpha, \beta \in \mathbb{R}_{>0}$ are unbounded. Now, ρ determines the relative importance between the likelihood and the total regularization imposed on both networks and γ weights the proportion of total regularization between the mean and precision networks. Here, $\rho = 1$ corresponds to the MLE objective while $\rho \to 0$ could be interpreted as converging to the mode of the prior in a Bayesian setting. Fixing $\gamma = 1$ leads to an unregularized precision function while choosing $\gamma = 0$ results in an unregularized mean function.

Description of Phases Model solutions across the ρ-γ hyperparameter space exhibit different traits. Similar to physical systems, this can be described as a collection of typical states or *phases* that make up a *phase diagram*. Fig. 1 shows an example phase diagram along with model fits coming from specific (ρ, γ) pairings. We argue that there are five primary regions of interest:

- Region U_{I}: Both the mean and precision functions are heavily regularized. The mean and standard deviation functions are fixed to the values they were initialized to.
- Region U_{II}: The mean function is heavily regularized and tends to be flat, as in Region U_I. Despite the constant mean function, the precision function still adapts to the data.

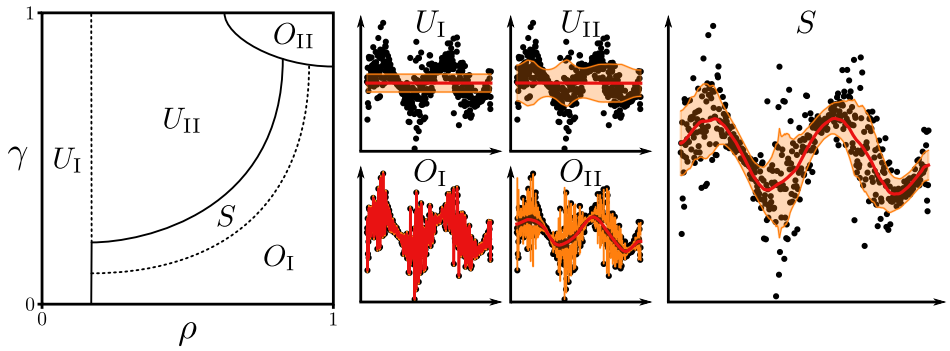

Figure 1: Visualization of a typical phase diagram in ρ-γ regularization space for a heteroskedastic regression model shown on left. Solid and dotted lines indicate sharp and smooth transitions in model behavior respectively. Example model mean fits shown in red (with pointwise $\pm$ standard deviation in orange) from the NFE for each key phase in middle and right plots.

• Region O_{I}: Heavily overfit mean and the residuals and standard deviations essentially vanish. Increasing $\rho \to 1$ yields true MLE fits (seen on the right).

• Region O_{II}: The mean function does not overfit due to regularization, leaving large residuals for the lowly regularized precision function to overfit onto.

• Region S: The mean function and standard deviations adapt to the data without overfitting. We conjecture that solutions in this region will provide the best generalization.

Nonparametric Modeling & Field Theory We propose abstracting away the neural networks $\hat{\mu}_\theta$ and $\hat{\Lambda}_\phi$ and instead analyzing nonparametric, twice-differentiable functions $\hat{\mu}$ and $\hat{\Lambda}$ respectively. Since these functions are nonparametric, we can no longer use L_2 penalties. A comparable substitute is to directly penalize the output "complexity" of the models, or the cumulative absolute rate of change: $\int \alpha ||\nabla \hat{\mu}(x)||_2^2 dx$ and $\int \beta ||\nabla \hat{\Lambda}(x)||_2^2 dx$. Note that these specific penalizations induce similar limiting behaviors for resulting solutions (i.e., $\alpha, \beta \to 0$ implies overfitting while $\to \infty$ implies constant functions).

Field theories are non-parametric descriptions of the spatial (or spatiotemporal) configurations of continuous physical systems (12). For example, the local magnetic field (magnetization) of a two or three-dimensional magnetic material would be an example of a *field*. For time independent problems we minimize a free energy functional: we will refer to it as *nonparametric free energy (NFE)*. The NFE depends on certain parameters, such as the strength of an external magnetic field or a temperature. Upon smoothly varying these parameters, the most likely field configuration can undergo smooth changes ("crossovers") or abrupt changes ("phase transitions"). As follows, we outline a field-theoretical treatment of the mean and variance parameters of our heteroskedastic regression model, where the mean and variance show phase transitions upon varying their regularization strengths.

Using the assumptions outlined above, the cross-entropy objective can be interpreted as an action functional of a corresponding two-dimensional NFE:

$$\begin{aligned}\mathcal{L}_{\rho,\gamma}(\hat{\mu},\hat{\Lambda}) = \int_{\mathcal{X}} p(x)\rho \int_{\mathcal{Y}} p(y\,|\,x)\left[\frac{1}{2}\hat{\Lambda}(x)\left(y-\hat{\mu}(x)\right)^2 - \frac{1}{2}\log\hat{\Lambda}(x)\right]dy \\ + (1-\rho)\left[\gamma||\nabla\hat{\mu}(x)||_2^2 + (1-\gamma)||\nabla\hat{\Lambda}(x)||_2^2\right]dx. \end{aligned} \tag{4}$$

The nested integral complicates matters so we consider the scenario in which the inner integral is approximated using a single MC sample. Numerically, we can arrive at approximate solutions for a given dataset. Analytically, we can gain insights into the solutions of the NFE by taking functional derivatives with respect to $\hat{\mu}$ and $\hat{\Lambda}$ and setting them to zero. We find the following conditions must be met for any set of solutions:

$$\begin{cases} \frac{\delta\mathcal{L}_{\rho,\gamma}(\hat{\mu},\hat{\Lambda})}{\delta\hat{\mu}} \triangleq 0 \\ \frac{\delta\mathcal{L}_{\rho,\gamma}(\hat{\mu},\hat{\Lambda})}{\delta\hat{\Lambda}} \triangleq 0 \end{cases} \implies \begin{cases} \hat{\Lambda}^*(x)(\hat{\mu}^*(x)-y(x)) &= 2\left(\frac{1-\rho}{\rho}\right)\gamma\frac{\Delta\hat{\mu}^*(x)}{p(x)} \\ (\hat{\mu}^*(x)-y(x))^2 &= \frac{1}{\hat{\Lambda}^*(x)} + 4\left(\frac{1-\rho}{\rho}\right)(1-\gamma)\frac{\Delta\hat{\Lambda}^*(x)}{p(x)}, \end{cases} \tag{5}$$

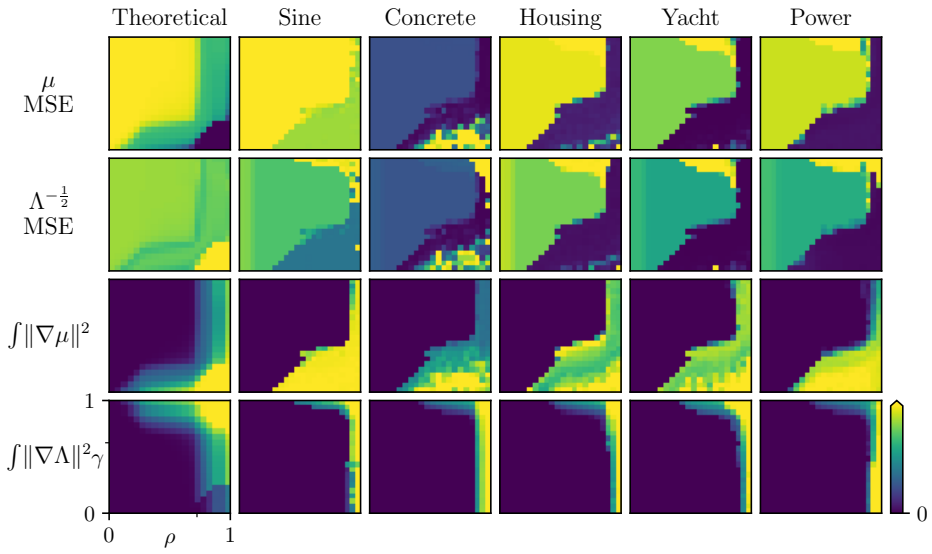

Figure 2: Grid of various metrics (rows) evaluated on different data or fitting techniques (columns). The left most column holds the results from the theoretical NFE; remaining columns show results on test data from fitted neural networks. The ticks on the lower left plot mark $\gamma = 0.5$ and $\rho = 0.5$.

where Δ is the Laplace operator (14). These equalities hold *almost everywhere* with respect to $p(x)$.

NFE Insights Under the assumptions of our NFE, the following properties hold: (i) in the absence of regularization ($\rho = 1$), there are no solutions to the NFE; (ii) in the absence of data ($\rho = 0$), there is no unique solution to the NFE; and (iii) there are no valid solutions to the NFE if $\rho \in (0, 1), \gamma = 1$. Should there be no mean regularization, then there needs to be at least some regularization for the precision. Proofs for all of these results are deferred to the Appendix A.2. Importantly, these limiting cases match our intuition for the solutions to the NFE (which were equivalent to the neural network setting). Empirically, we have discovered that the phase diagram typically resembles Fig. 1.

3 Experiments

We analyze the effects of regularization on several one-dimensional simulated datasets and standardized versions of the *Concrete* (15), *Housing* (16), *Power* (17), and *Yacht* (18) regression datasets from the UCI Machine Learning Repository (19). We fit neural networks to the simulated and real-world data and additionally solve our NFE for the simulated data. Detailed descriptions of the data, training details, and complete results, are included in Appendix B.1, Appendix B.2, and Appendix B.4.

The *Dirichlet energy*, $\int_{\mathcal{X}} ||\nabla f(x)||_2^2 \, dx$, captures how expressive a function is, with more expressive functions yielding higher values. We also consider the mean squared error (MSE) between predicted mean $\hat{\mu}_\theta(x_i)$ and target y_i, as well as between predicted standard deviation ($\Lambda^{-\frac{1}{2}}(x_i)$) and absolute residual $|\hat{\mu}_\theta(x_i) - y_i|$. Low values indicate the mean and standard deviation fit the data well.

We present summaries of the fitted models in grids with ρ on the x-axis and γ on the y-axis in Fig. 2. The far right column ($\gamma = 1$) corresponds to MLE solutions. The main focus is on qualitative traits of fits under different levels of regularization and how they behave in a relative sense. Our metrics show sharp phase transitions upon varying ρ, γ, as in a physical system. The NFE insights and observed phases are consistent with the NFE and results from fitting neural networks. Thus, our results are not tied to a specific architecture or datasest. Finally, our phase diagrams suggest that we can search along $\rho = 1 - \gamma$ to find a well-calibrated (ρ, γ)-pair from region S. We discuss this further in Appendix Appendix C.2.

4 Conclusion

We used field-theoretical tools from statistical physics to derive an NFE, which allowed us to produce analytical insights into the pathologies of deep heteroskedastic regression. These insights generalize across models and datasets and provide a theoretical explanation for the need for carefully tuned regularization in these models, due to sharp phase transitions between pathological solutions. We also presented a numerical approximation to this theory, which empirically agrees with neural network solutions to synthetic and real-world data. Using insights from the theory, we have shown that we can tune the required regularization for these models more efficiently than would naïvely be the case.

Acknowledgements Eliot Wong-Toi acknowledges support from the Hasso Plattner Research School at UC Irvine. Alex Boyd acknowledges support from the National Science Foundation Graduate Research Fellowship grant DGE-1839285. Vincent Fortuin was supported by a Branco Weiss Fellowship. Stephan Mandt acknowledges support by the IARPA WRIVA program, the National Science Foundation (NSF) under the NSF CAREER Award 2047418; NSF Grants 2003237 and 2007719, the Department of Energy, Office of Science under grant DE-SC0022331, as well as gifts from Intel, Disney, and Qualcomm.

References

[1] A. Mathew, P. Amudha, and S. Sivakumari, "Deep Learning Techniques: An Overview," in *Advanced Machine Learning Technologies and Applications* (A. E. Hassanien, R. Bhatnagar, and A. Darwish, eds.), Advances in Intelligent Systems and Computing, (Singapore), pp. 599–608, Springer, 2021.

[2] J. Ahmad, H. Farman, and Z. Jan, "Deep Learning Methods and Applications," in *Deep Learning: Convergence to Big Data Analytics* (M. Khan, B. Jan, and H. Farman, eds.), SpringerBriefs in Computer Science, pp. 31–42, Singapore: Springer, 2019.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[4] N. Skafte, M. Jørgensen, and S. Hauberg, "Reliable training and estimation of variance networks," 2019.

[5] V. Fortuin, M. Collier, F. Wenzel, J. Allingham, J. Liu, D. Tran, B. Lakshminarayanan, J. Berent, R. Jenatton, and E. Kokiopoulou, "Deep Classifiers with Label Noise Modeling and Distance Awareness," *Transactions on Machine Learning Research*, 2022.

[6] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Neural Information Processing Systems*, 2017.

[7] D. Nix and A. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, pp. 55–60 vol.1, June 1994.

[8] A. Stirn and D. A. Knowles, "Variational Variance: Simple, Reliable, Calibrated Heteroscedastic Noise Variance Parameterization," Oct. 2020. arXiv:2006.04910 [cs, stat].

[9] M. Seitzer, A. Tavakoli, D. Antic, and G. Martius, "ON THE PITFALLS OF HETEROSCEDASTIC UNCERTAINTY ESTIMATION WITH PROBABILISTIC NEURAL NETWORKS," 2022.

[10] A. Stirn, H.-H. Wessels, M. Schertzer, L. Pereira, N. E. Sanjana, and D. A. Knowles, "Faithful Heteroscedastic Regression with Neural Networks," 2023.

[11] L. D. Landau and E. M. Lifshitz, *Statistical Physics: Volume 5*. Elsevier, Oct. 2013.

[12] A. Altland and B. D. Simons, *Condensed Matter Field Theory*. Cambridge: Cambridge University Press, 2 ed., 2010.

[13] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, pp. 251–257, Jan. 1991.

[14] E. Engel and R. M. Dreizler, *Density Functional Theory: An Advanced Course*. Theoretical and Mathematical Physics, Berlin, Heidelberg: Springer, 2011.

[15] I.-C. Yeh, "Concrete Compressive Strength," 2007.

[16] D. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of Environmental Economics and Management*, vol. 5, pp. 81–102, Mar. 1978.

[17] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 126–140, Sept. 2014.

[18] J. Gerritsma, "Geometry, resistance and stability of the Delft Systematic Yacht hull series," *TU Delft, Faculty of Marine Technology, Ship Hydromechanics Laboratory, Report No. 520-P, Published in: International Shipbuilding Progress, ISP, Delft, The Netherlands, Volume 28, No. 328, also 7th HISWA Symposium, Amsterdam, The Netherlands*, 1981.

[19] M. Kelly, R. Longjohn, and K. Nottingham, "The UCI Machine Learning Repository."

A Theoretical Details

A.1 Full Functional Derivatives

Full functional derivatives of our NFE are:

$$\mathcal{L}_{\rho,\gamma}(\hat{\mu},\hat{\Lambda}) \approx \int_{\mathcal{X}} p(x)\rho\left[\frac{1}{2}\hat{\Lambda}(x)\left(y(x)-\hat{\mu}(x)\right)^2 - \frac{1}{2}\log\hat{\Lambda}(x)\right] + (1-\rho)\left[\gamma||\nabla\hat{\mu}(x)||_2^2 + (1-\gamma)||\nabla\hat{\Lambda}(x)||_2^2\right]dx$$

$$\begin{cases} \frac{\delta\mathcal{L}}{\delta\hat{\mu}} & = p(x)\rho\hat{\Lambda}(x)(\hat{\mu}(x)-y(x)) - 2(1-\rho)\gamma\Delta\hat{\mu}(x) \\ \frac{\delta\mathcal{L}}{\delta\hat{\Lambda}} & = \frac{p(x)\rho}{2}\left[(y(x)-\hat{\mu}(x))^2 - \frac{1}{\hat{\Lambda}(x)}\right] - 2(1-\rho)(1-\gamma)\Delta\hat{\Lambda}(x) \end{cases} \tag{6}$$

After setting equal to zero we arrive at

$$\begin{cases} \hat{\Lambda}^*(x)(\hat{\mu}^*(x)-y(x)) = 2\left(\frac{1-\rho}{\rho}\right)\gamma\frac{\Delta\hat{\mu}^*(x)}{p(x)} \\ (y(x)-\hat{\mu}^*(x))^2 = \frac{1}{\hat{\Lambda}^*(x)} + 4\left(\frac{1-\rho}{\rho}\right)(1-\gamma)\frac{\Delta\hat{\Lambda}^*(x)}{p(x)} \end{cases} \tag{7}$$

A.2 Proofs

Proposition 1. *Assuming there exists twice differentiable functions $\mu : \mathbb{R}^d \to \mathbb{R}, \Lambda : \mathbb{R}^d \to \mathbb{R}_{>0}$, the following properties hold*

- *i In the absence of regularization ($\rho = 1$), there are no solutions to the NFE.*
- *ii In the absence of data ($\rho = 0$), there is no unique solution to the NFE.*
- *iii There are no valid solutions to the NFE if $\rho \in (0,1), \gamma = 1$. Some regularization on precision function is needed for a solution to potentially exist.*

Proof. Without loss of generality, we consider a uniform $p(x)$ and drop it from the equations.

(i) When $\rho = 1$ the necessary conditions for an optima are

$$\begin{cases} \hat{\Lambda}^*(x)(\hat{\mu}^*(x)-y(x)) = 0 \\ (\hat{\mu}^*(x)-y(x))^2 = \frac{1}{\hat{\Lambda}^*(x)} \end{cases} \tag{8}$$

$$\implies \begin{cases} \hat{\Lambda}^*(x)(\hat{\mu}^*(x)-y(x)) = 0 \\ \hat{\Lambda}^*(x)(\hat{\mu}^*(x)-y(x))^2 = 1 \end{cases} \tag{9}$$

$$\implies \begin{cases} \hat{\Lambda}^*(x)(\hat{\mu}^*(x)-y(x)) = 0 \\ 0 \times (\hat{\mu}^*(x)-y(x)) = 1 \end{cases} \tag{10}$$

$$\implies 0 = 1 \tag{11}$$

which is a contradiction and there cannot exist μ, Λ that are solutions.

(ii) When $\rho = 0$ the integral we seek to maximize is:

$$\mathcal{L}_{\rho,\gamma}(\hat{\mu},\hat{\Lambda}) = \int_{\mathcal{X}} \rho \int_{\mathcal{Y}} p(y\,|\,x)\left[\frac{1}{2}\hat{\Lambda}(x)\left(y-\hat{\mu}(x)\right)^2 - \frac{1}{2}\log\hat{\Lambda}(x)\right]dy + (1-\rho)\left[\gamma||\nabla\hat{\mu}(x)||_2^2 + (1-\gamma)||\nabla\hat{\Lambda}(x)||_2^2\right]dx \tag{12}$$

$$= \int_{\mathcal{X}} \left[\gamma||\nabla\hat{\mu}(x)||_2^2 + (1-\gamma)||\nabla\hat{\Lambda}(x)||_2^2\right]dx. \tag{13}$$

Each term in this integral is non-negative, so the minimum value it could be is zero. Any pair of constant functions μ, Λ will minimize this integral, of which there are infinitely many.

(iii) We return to the α, β-parameterization for this proof. Suppose there is no mean regularization, that is $\alpha = 0$.

$$\implies \begin{cases} \hat{\Lambda}^*(x)(\hat{\mu}^*(x) - y(x)) = 0 \\ (\hat{\mu}^*(x) - y(x))^2 = \frac{1}{\hat{\Lambda}^*(x)} + 4\beta\Delta\hat{\Lambda}^*(x) \end{cases} \tag{14}$$

From the first condition we see that there must be perfect matching between $\hat{\mu}^*$ and y since $\hat{\Lambda}^* > 0$ in order to define a valid normal distribution.

$$\implies \begin{cases} (\hat{\mu}^*(x) - y(x)) = 0 \\ (\hat{\mu}^*(x) - y(x))^2 = \frac{1}{\hat{\Lambda}^*(x)} + 4\beta\Delta\hat{\Lambda}^*(x) \end{cases} \tag{15}$$

$$\implies \begin{cases} (\hat{\mu}^*(x) - y(x)) = 0 \\ 0 = \frac{1}{\hat{\Lambda}^*(x)} + 4\beta\Delta\hat{\Lambda}^*(x) \end{cases} \tag{16}$$

Now, note that as $\beta \to 0$,

$$\implies \begin{cases} (\hat{\mu}^*(x) - y(x)) = 0 \\ 0 = \frac{1}{\hat{\Lambda}^*(x)} \end{cases} \tag{17}$$

but $\hat{\Lambda}^* \in \mathbb{R}$, so the second condition can never be satisfied. Thus, in order for a solution to exist if $\alpha = 0 \implies \beta > 0$. These α, β values correspond to $\rho \in (0, 1), \gamma \neq 1$.

□

This proposition implies the existence, or rather the lack thereof of solutions to the NFE. Should there be no mean regularization, then there needs to be at least some present for the precision. The theory potentially suggests that the vice versa of this should also guarantee valid solutions (i.e., $\alpha > 0$ and $\beta = 0$); however, in practice this does not hold true. The reason lies in the stipulation that $y(x) \neq \hat{\mu}(x)$ a.e.

Typically, this condition can be satisfied while still allowing for countably many values of x in which $y(x) = \hat{\mu}(x)$. The problem is that we fit models using a finite amount of data. As mentioned previously, we typically minimize the objective function by approximating it using a MC estimate with $\mathcal{D}$ as samples. An alternative perspective of this decision is that we are actually calculating the expected values exactly with respect to an empirical distribution imposed by $\mathcal{D}$: $p(x, y) :\propto \sum_{(x_i, y_i) \in \mathcal{D}} \delta(||x - x_i||)\delta(y - y_i)$ where $\delta(\cdot)$ is the Dirac delta function.[1] Because of this, a single value of x can possess non-zero measure, thus it only takes a single instance of $y(x) = \hat{\mu}(x)$ for the statement $y(x) \neq \hat{\mu}(x)$ a.e. to be false. This, unfortunately, is very likely to happen while solving for $\hat{\mu}, \hat{\Lambda}$. **Thus, we can conclude that no matter what, $\left(\frac{1-\rho}{\rho}\right)(1 - \gamma) > 0$ for a valid solution to be guaranteed to exist.**

B Experimental Details

B.1 Datasets

We chose 64 datapoints in each of the simulated datasets. The generating processes for each simulated dataset is included in Table 1 and can be seen in Fig. 3. The homoskedastic data is simulated in the same way, but with $f(x) = 1$. For testing, we simulate a new dataset of 64 datapoints with the same process. Table 2 summarizes the UCI datasets.

B.2 Training Details

We take 22 values of γ, ρ that range from 10^{-10} up to $1 - 10^{-5}$ ($\rho, \gamma \in \{0.9999, 0.999, 0.99, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 0.00000001, 0.000000001, 0.0000000001, 0.00000000001\}$)

[1] Not to be confused with the functional derivative operator δ.

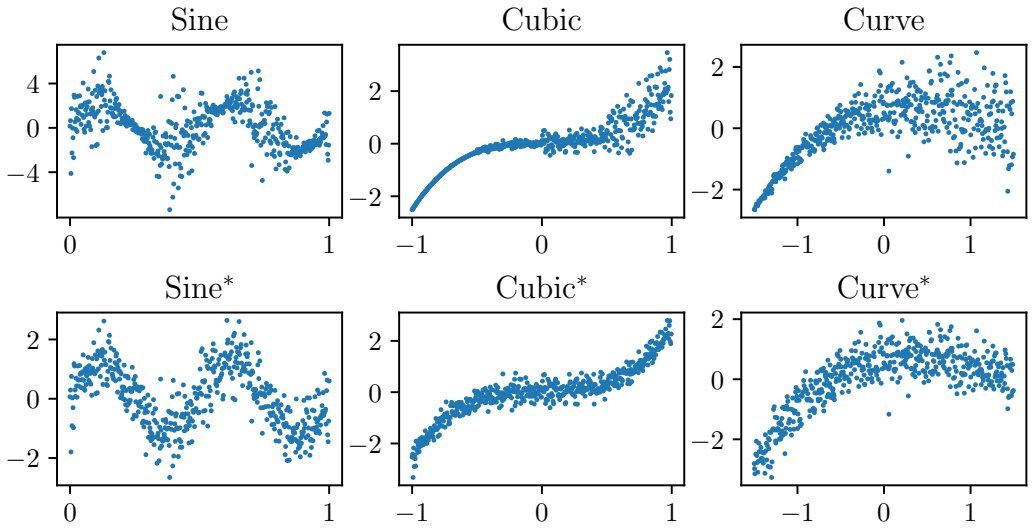

Figure 3: Visualization of heteroskedastic and homoskedastic versions of simulated datasets. Specific details for the functional form of these can be found in Table 1.

Table 1: Simulated datasets. Each dataset is defined by a true μ function and then a noise function f. All data is generated as $\mu(x) + \epsilon(x)$ where $\epsilon(x) \sim \mathcal{N}(0, f(x)^2)$. After the datasets were generated they were scaled to have mean zero and standard deviation one. The homoskedastic versions of each dataset fix $f(x) = 1$. The datasets are shown in Fig. 3.

Dataset	Mean (μ)	Noise Pattern (f)	Domain
Sine	$\mu(x) = 2\sin(4\pi x)$	$f(x) = \sin(6\pi x) + 1.25$	$x \in [0, 1]$
Cubic	$\mu(x) = x^3$	$f(x) = \begin{cases} 0.1 & \text{for } x < -0.5 \\ 1 & \text{for } x \in [-0.5, 0.0) \\ 3 & \text{for } x \in [0.0, 0.5) \\ 10 & \text{for } x \geq .5 \end{cases}$	$x \in [-1, 1]$
Curve	$\mu(x) = x - 2x^2 + 0.5x^3$	$f(x) = x + 1.5$	$x \in [-1.5, 1.5]$

Table 2: UCI dataset.

Dataset	Train Size	Test Size	Input Dimension
Concrete	687	343	8
Housing	337	168	13
Power	6379	3189	4
Yacht	204	102	6

on a logit scale for all of the experiments run on neural networks. For the NFEs we take 20 values from 10^{-6} up to $1 - 10^{-7}$ ($\rho, \gamma \in \{0.999999, 0.99999, 0.99999, 0.9999, 0.999, 0.99, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$), also on a logit scale. This scaling increases the absolute density of points evaluated near the extreme cases of 0 and 1 where the theoretical analysis of the NFE focused. The ranges differ slightly due to numerical stability during the fitting. The limiting cases of $\gamma, \rho \in \{0, 1\}$ were omitted for numerical stability and the ranges of values for the NFEs vs neural networks vary slightly for the same reason. All experiments were run on Nvidia Quadro RTX 8000 GPUs. Approximately 400 total GPU hours were used across all experiments.

B.3 NFE

For the discretized field theory we take $n_{ft} = 4096$ evenly spaced points on the interval $[-1, 1]$. There are two datapoints placed beyond $[-1, 1]$ because the method we use to estimate the gradients requires the datapoints to have left and right neighbors. These datapoints were not included when computing our metrics. Of these 4096 datapoints 64 were randomly selected to be used for training neural networks $\hat{\mu}_\theta, \hat{\Lambda}_\phi$. The field theory results were consistent across choices of $n_{ft} \in \{256, 512, 1024, 2048, 4096\}$. We present results for $n_{ft} = 4096$ in the main paper. We train for 100000 epochs and use the Adam optimizer with a basic triangular cycle that scales initial amplitude by half each cycle on the learning rate. The minimum and maximum learning rates were 0.0005 and 0.01. The cycles were 5000 epochs long. We clip the gradients at 1000.

B.4 Simulated Data with Neural Networks

For all of the simulated datasets except for *sine* we train for 600000 epochs and use the Adam optimizer with a basic triangular cycle that scales initial amplitude by half each cycle on the learning rate. The minimum and maximum learning rates were 0.0001 and 0.01. The cycles were 50000 epochs. The first 250000 epochs are only spend on training $\hat{\mu}_\theta$ while the remaining 350000 epochs are spent training both $\hat{\mu}_\theta, \hat{\Lambda}_\phi$. We clip the gradients at 1000. The training for the *sine* dataset was the same, except trained for 2500000 epochs.

B.5 UCI Data with Neural Networks

For the *concrete, housing* and *yacht* datasets we train for 500000 epochs and use the Adam optimizer with a basic triangular cycle that scales initial amplitude by half each cycle on the learning rate. The minimum and maximum learning rates were 0.0001 and 0.01. The cycles were 50000 epochs. The first 250000 epochs are only spend on training $\hat{\mu}_\theta$ while the remaining 250000 epochs are spent training both $\hat{\mu}_\theta, \hat{\Lambda}_\phi$. Meanwhile on the *power* dataset, we had to use minibatching due to the size of the dataset. We used minibatches of 1000 and trained for 50000 total epochs with the first 25000 dedicated solely to $\hat{\mu}_\theta$ and the remainder training both $\hat{\mu}_\theta, \hat{\Lambda}_\phi$. The same cyclic learning rate was used but with cycle length 5000. We clip the gradients at 1000.

C Additional Results

C.1 All Synthetic Dataset Results

Both NFE and neural networks were fit to the heteroskedastic and homoskedastic synthetic datasets described in Table 1. The main results for these displayed as phase diagrams of various metrics can be seen in Fig. 4 and Fig. 5 respectively. We largely see the same trends as were exhibited by the real-world datasets seen in Fig. 2.

C.2 Diagonal Hyperparameter Search

We perform a search along the $\rho = 1 - \gamma$ minor diagonal in the phase diagrams. Fig. 6 shows the summary statistics along the slice where $\rho = 1 - \gamma$. After conducting our diagonal search we found the model that minimized μ MSE and the model that minimized $\Lambda^{-\frac{1}{2}}$ MSE on the *training* data. In some cases these models coincided. We then used the model that was on the midpoint (on a logit scale) of the $\rho + \gamma = 1$ line between these two models to compare against baselines from (9) and

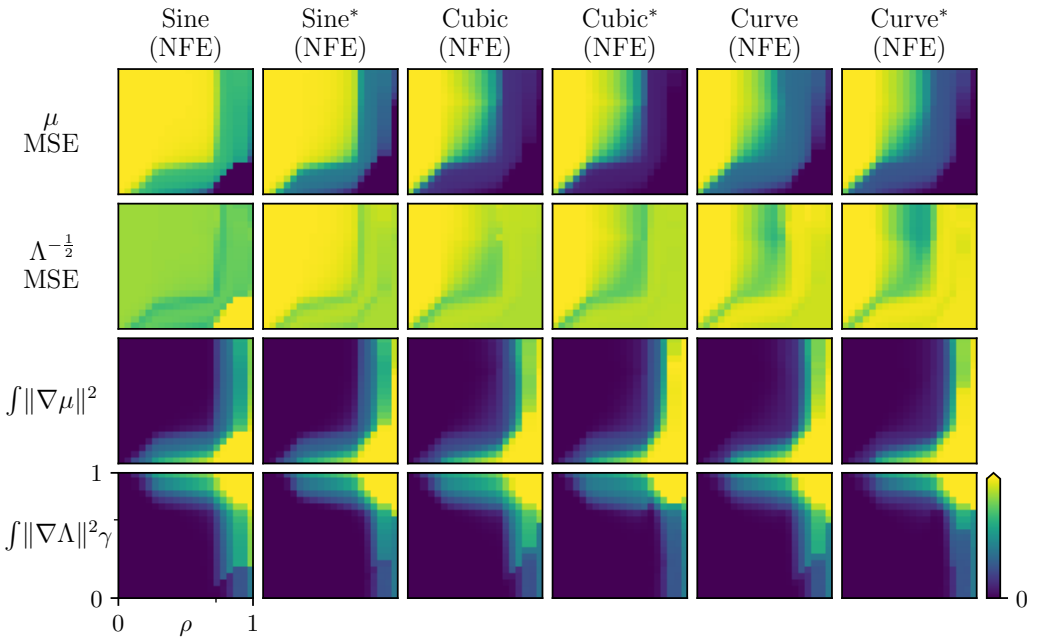

Figure 4: Same configuration as Fig. 2, except all results here pertain to minimizing the NFE on six different synthetic datasets described in Table 1. Dataset names with an $*$ are the homoskedastic counterparts.

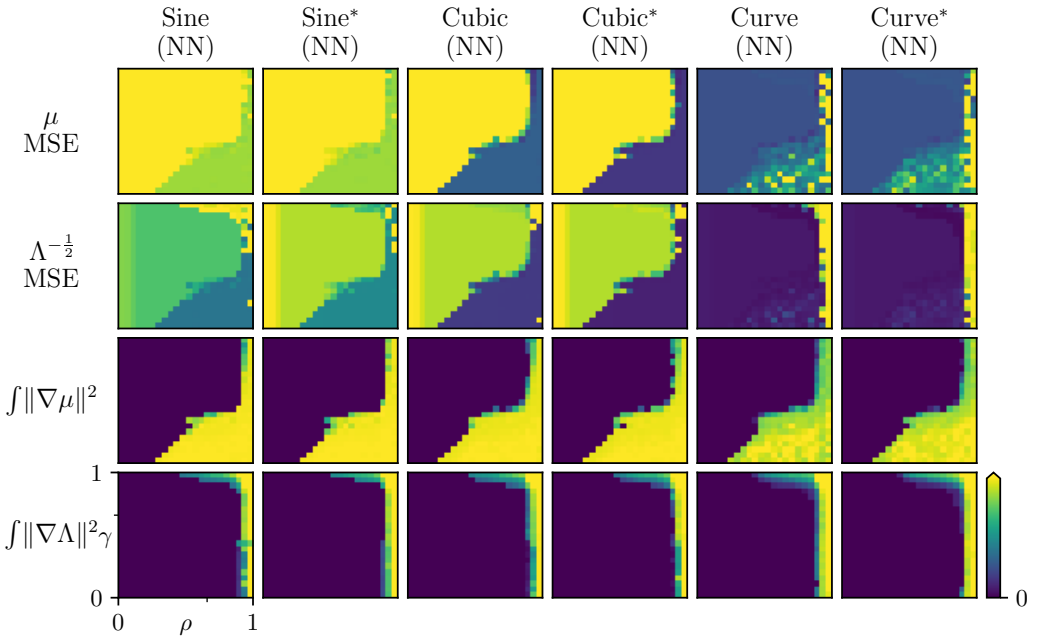

Figure 5: Same configuration as Fig. 2 and Fig. 4, except all results here pertain to training a neural network on six different synthetic datasets described in Table 1. Dataset names with an $*$ are the homoskedastic counterparts.

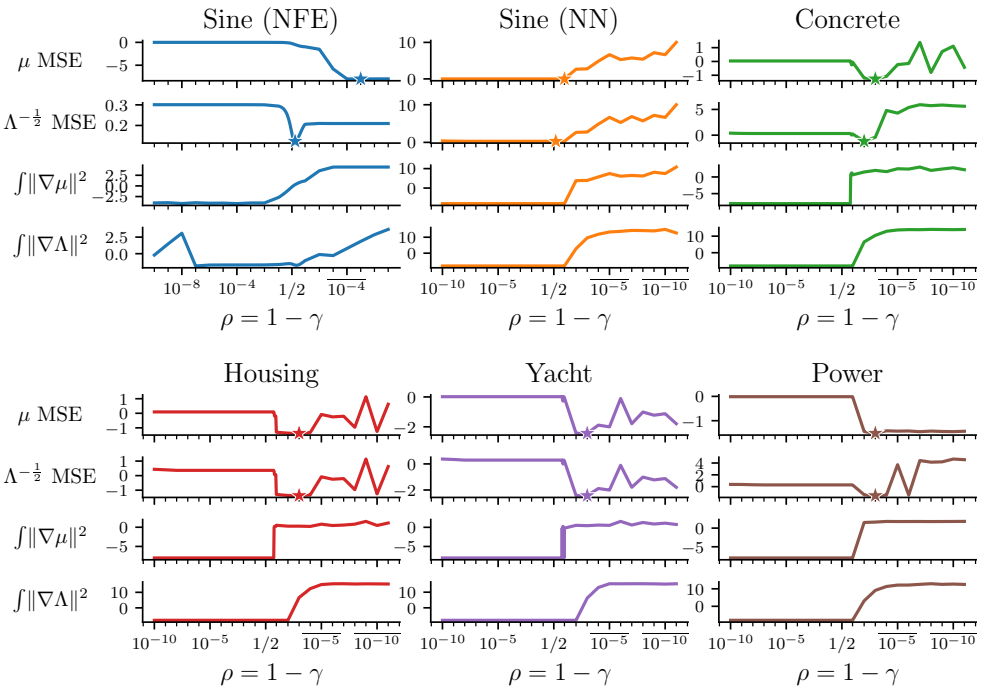

Figure 6: Test metrics for six different settings achieved with varying values of $\rho \in (0, 1)$ and with γ restricted to be equal to $1 - \rho$. Stars indicate minimum MSE values. All metrics are reported on a $\log_{10}$ scale. ρ values are shown on a logit scale with $\overline{10^k} := 1 - 10^k$. From left to right, note the sharp decrease in test metric values, especially in the solutions to neural network models followed by a typical smoother increase. This empirically supports the existence of the well-calibrated S phase shown in Fig. 1 and allows for hyperparameter optimization in $O(N)$ instead of $O(N^2)$.

(6). The results are reported in Table 3. In all cases our method is competitive with or exceeds the performance of these two baselines–particularly on real-world data.

Table 3: Comparison of our method against two baselines. We report the average and standard deviations of expected calibration error (ECE), μ MSE and $\Lambda^{-\frac{1}{2}}$ on test data. Lowest mean value for each metric is bolded.

Dataset Metric	**Ours**	β-NLL (9)	MLE Ensemble (6)
Cubic			
μ MSE	0.2339 ± 0.01	**0.1500** ± 0.01	1.1809 ± 1.88
$\Lambda^{-\frac{1}{2}}$MSE	0.2397 ± 0.02	**0.1397** ± 0.01	inf ± nan
Curve			
μ MSE	**0.4318** ± 0.12	0.4877 ± 0.16	1.0067 ± 0.19
$\Lambda^{-\frac{1}{2}}$MSE	0.4655 ± 0.09	**0.4187** ± 0.20	inf ± nan
Sine			
μ MSE	**0.7968** ± 0.00	4.4107 ± 6.90	0.9716 ± 0.06
$\Lambda^{-\frac{1}{2}}$MSE	**0.7968** ± 0.00	4.3524 ± 6.89	inf ± nan
Concrete			
μ MSE	**0.1055** ± 0.02	14.9882 ± 28.75	2.2454 ± 1.74
$\Lambda^{-\frac{1}{2}}$MSE	**0.3028** ± 0.51	$1.3 \times 10^5 \pm 2.7 \times 10^5$	$1.3 \times 10^5 \pm 1.2 \times 10^5$
Housing			
μ MSE	**1.2236** ± 0.00	851.8968 ± 1985.56	155.4494 ± 128.27
$\Lambda^{-\frac{1}{2}}$MSE	**0.7610** ± 0.00	851.8959 ± 1985.56	218.8269 ± 195.38
Power			
μ MSE	0.0350 ± 0.01	0.0313 ± 0.01	**0.0177** ± 0.00
$\Lambda^{-\frac{1}{2}}$MSE	0.0343 ± 0.01	0.0360 ± 0.01	**0.0091** ± 0.00
Yacht			
μ MSE	**0.0077** ± 0.01	34.1239 ± 194.87	6.2670 ± 13.96
$\Lambda^{-\frac{1}{2}}$MSE	**0.0076** ± 0.01	34.1237 ± 194.87	8.0599 ± 19.18