5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–16

Online Laplace Model Selection Revisited

Anonymous Authors
Anonymous Institution

Abstract

The Laplace approximation provides a closed form model selection objective for neural networks (NN). *Online* variants, which optimise NN parameters jointly with hyperparameters, like weight decay strength, have seen renewed interest in the Bayesian deep learning community. However, these methods violate Laplace's method's critical assumption that the approximation is performed around a mode of the loss, calling into question their soundness. This work re-derives online Laplace methods, showing them to target a variational bound on a mode-corrected variant of the Laplace evidence which does make stationarity assumptions. Online Laplace and its mode-corrected counterpart share stationary points where 1. the NN parameters are a *maximum a posteriori*, satisfying the Laplace method's assumption, and 2. the hyperparameters maximise the Laplace evidence, motivating online methods. We demonstrate that these optima are roughly attained in practise by online algorithms using full-batch gradient descent on UCI regression datasets. The optimised hyperparameters prevent overfitting and outperform validation-based early stopping.

1. Introduction

Online model selection holds the promise of automatically tuning large numbers of neural network (NN) hyperparameters during a single training run, obsoleting cross validation (Clarke et al., 2022). Online Laplace methods (OL) interleave regular NN optimisation steps with steps of hyperparameter optimisation with respect to the Laplace-approximated model evidence (Foresee and Hagan, 1997; Friston et al., 2007; Immer et al., 2021a). The method has been used to learn layer-wise weight decay strengths, data noise scales, and even data augmentation hyperparameters (Immer et al., 2022).

Laplace's method constructs a second-order Taylor expansion around a mode of the posterior. Here, the first-order term vanishes, leaving a pure quadratic approximation to the log-density, that is, a Gaussian. The Laplace evidence is this distribution's normalising constant. The online setting implies non-convergence of the NN parameters, violating the stationarity assumption. Antorán et al. (2022b) show that applying the Laplace method to pre-trained NNs which do not attain zero loss gradient leads to severe deterioration in model selection performance. Immer et al. (2021a) experiment with inclusion of the first order Taylor term in the OL procedure but find it leads to training instability.

This work revisits online Laplace methods, taking steps towards reconciling their seeming unsoundness with their satisfactory empirical performance. Our contributions are:

- We show that a variant of OL that includes the first-order Taylor term corresponds to NN hyperparameter optimisation with the evidence of a tangent linear model. At this procedure's fixed points, the linear model's *maximum a posteriori* (MAP) parameters match the NN parameters, for which they are also a MAP and the hyperparameters optimise the linear model's evidence, which in turn matches the NN's Laplace evidence.

© A. Authors.

- We re-derive the standard OL objective as a variational lower bound on the evidence of the tangent linear model described above. OL shares fixed points, where the ELBO is tight, with the above first-order-corrected procedure, motivating its use.
- We show that these fixed points are roughly attained in practise by small NNs trained via full-batch gradient descent with online hyperparameter tuning on UCI regression datasets, but they are not attained without online tuning. The hyperparameters found through online Laplace lead to NNs that outperform their non-online counterparts.

2. Preliminaries: Laplace approximation and online variants

We consider a supervised problem where n inputs, stacked as $X \in \mathbb{R}^{n \times d_x}$, are paired with scalar targets stacked as $y \in \mathbb{R}^n$. We introduce a NN regressor $f : \mathbb{R}^{d_w} \times \mathbb{R}^{d_x} \to \mathbb{R}$ with $w \in \mathbb{R}^{d_w}$ as parameters. Omitting dependence on the inputs from our notation, as these are fixed, and working with arrays of stacked predictions $f(w) \in \mathbb{R}^n$, we model the targets as

$$y = f(w) + \epsilon \quad \text{with} \quad w \sim \mathcal{N}(0, \alpha^{-1} I_{d_w}) \quad \text{and} \quad \epsilon \sim \mathcal{N}(0, \beta^{-1} I_n),$$

where α is the scalar precision of our prior over NN parameters and β is the scalar homoscedastic noise precision. The set of both hyperparameters is $\mathcal{M} = \{\alpha, \beta\}$. Our results readily generalise to non-scalar, multi-output settings (Antoran et al., 2023).

We train the NN to minimise the loss $\ell_f(w; \mathcal{M}) = \frac{\beta}{2}\|y - f(w)\|^2 + \frac{\alpha}{2}\|w\|^2$. This objective matches the unnormalised, negative log-joint density of our parameters and targets

$$e^{-\ell_f(w;\mathcal{M})} \left(\frac{\beta}{2\pi}\right)^{\frac{n}{2}} \left(\frac{\alpha}{2\pi}\right)^{\frac{d_w}{2}} = \mathcal{N}(y;\, f(w), \beta^{-1} I_n)\mathcal{N}(w;\, 0, \alpha^{-1} I_{d_w}) = p_f(y, w; \mathcal{M}).$$

Laplace's method (Mackay, 1992; Bishop, 2006) takes a second-order Taylor expansion of $\log p_f(y, w; \mathcal{M})$ around w_0, assumed to be a mode ($w_0 \in \arg\min_w \ell_f(w; \mathcal{M}) := w^\star$), and approximates the NN model evidence $p_f(y; \mathcal{M})$ by integrating the expanded expression as

$$\begin{aligned} &\int \exp\left(\log p_f(y, w_0; \mathcal{M}) - \frac{1}{2}(w - w_0)^\top H (w - w_0)\right) \mathrm{d}w \\ &= \exp\left(\frac{d_w}{2}\log\alpha + \frac{n}{2}\log\beta - \frac{\beta}{2}\|y - f(w_0)\|_2^2 - \frac{\alpha}{2}\|w_0\|_2^2 - \frac{1}{2}\log\det H - \frac{n}{2}\log 2\pi\right) \qquad (1) \\ &:= \exp \mathcal{L}_f(\mathcal{M}; w_0), \end{aligned}$$

where the first-order term vanishes since we assume $\partial_w \ell_f(w_0; \mathcal{M}) = 0$ and H is $\partial_w^2 \ell_f(w_0; \mathcal{M})$. In practice, the loss Hessian is replaced by the Generalised Gauss Newton matrix (GGN), that is, $H = \beta J^T J + \alpha I_{d_w} \in \mathbb{R}^{d_w \times d_w}$ for $J = \partial_w f(w_0) \in \mathbb{R}^{n \times d_w}$ the NN's Jacobian evaluated at the training inputs. This choice provides ease of computation and numerical stability (Martens, 2020; Immer et al., 2021b). We hereon assume H to be the GGN and suppress its dependence on the hyperparameters and the expansion point from our notation.

2.1. Online hyperparameter tuning

We may tune the hyperparameters $\mathcal{M}$ for a fully-optimised NN by choosing them to maximise $\mathcal{L}_f(\mathcal{M}; w^\star)$. This will not change our NN's predictions however, as its parameters are held

fixed at $w_\star$. Mackay (1992) proposes to re-train the NN from scratch using the updated hyperparameters. This procedure terminates when a joint stationary point of the parameters and hyperparameters is found: $w_\star \in \arg\min_w \ell_f(w; \mathcal{M}^\star)$ and $\mathcal{M}^\star \in \arg\min_{\mathcal{M}} \mathcal{L}_f(\mathcal{M}; w^\star)$.

The size of NN parameter spaces and datasets has grown dramatically since 1992 and re-training the NN is now impractical. This motivates online Laplace (OL) approaches which, at timestep t, perform a step of NN parameter w optimisation, minimising $\ell_f(w; \mathcal{M})$, followed by a hyperparameter update to maximise $\mathcal{L}_f(\mathcal{M}; w_t)$ (Foresee and Hagan, 1997; Friston et al., 2007; Immer et al., 2021a)[1]. Importantly, the Taylor expansion point is chosen to match the current NN parameter setting w_t. Since optimisation has not converged, $w_t \notin \arg\min_w \ell_f(w; \mathcal{M})$. Thus, $\mathcal{L}_f(\mathcal{M}; w_t)$, which discards the first-order expansion term, *does not provide a local approximation to the evidence.* Figure 1 illustrates that this first-order term can result in a vastly different approximation.

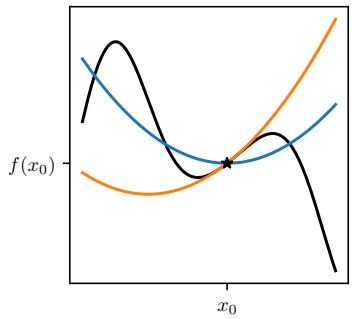

Figure 1: Second-order Taylor approximations of a function f (black) around x_0 (star), with (orange) and without (blue) the first-order term.

3. Understanding online Laplace through the tangent model

This section re-derives online Laplace methods for NNs as Bayesian inference in a tangent linear model. This analysis provides a non-heuristic motivation for these methods.

Online hyperparameter optimisation including the first-order term To start, we integrate the Taylor expansion of $\log p_f(y, w; \mathcal{M})$ at w_t without discarding the first-order term

$$\int \exp\left(\log p_f(y, w_t; \mathcal{M}) - \partial_w \ell_f(w_t; \mathcal{M})(w - w_t) - \frac{1}{2}(w - w_t)^\top H(w - w_t)\right) \mathrm{d}w \tag{2}$$
$$= \exp\left(\frac{d_w}{2}\log\alpha + \frac{n}{2}\log\beta - \frac{\beta}{2}\|y - (f(w_t) + J(v^\star - w_t))\|_2^2 - \frac{\alpha}{2}\|v^\star\|_2^2 - \frac{1}{2}\log\det H - \frac{n}{2}\log 2\pi\right)$$
$$:= \exp \mathcal{L}_h(\mathcal{M}; w_t),$$

where the optimum of the quadratic expansion $v^\star$ is obtained by performing a Gauss-Newton update on the NN loss at w_t, that is, $v^\star = w_t - \beta H^{-1}\left(J^T(f(w_t) - y) + \alpha w_t\right)$. Gauss-Newton optimisation is common in the NN second order optimisation-literature; it makes more progress per step than gradient descent (Botev et al., 2017; Martens, 2020; Barbano et al., 2023). Since $\mathcal{L}_h(\mathcal{M}; w_t)$ represents a local evidence approximation, mode-corrected by a Gauss-Newton step, it may more closely match the Laplace evidence computed at an optimum $\mathcal{L}_f(\mathcal{M}; w^\star)$ than the Laplace evidence computed at the online expansion point $\mathcal{L}_f(\mathcal{M}; w_t)$. We may use $\mathcal{L}_h(\mathcal{M}; w_t)$ to construct an algorithm where each step of NN parameter optimisation minimising $\ell_f(w; \mathcal{M})$ is interleaved with a hyperparameter update maximising $\mathcal{L}_h(\mathcal{M}; w_t)$.

1. Friston et al. (2007) refers to the described online Laplace procedure as *Variational Laplace*.

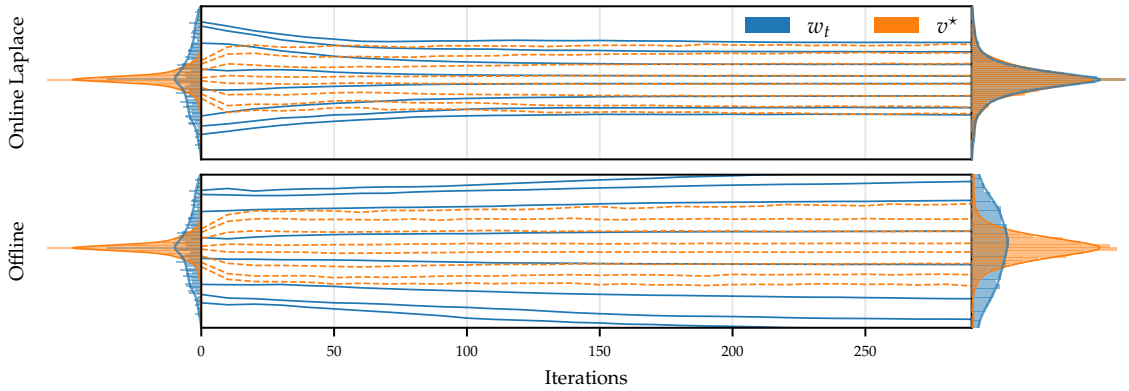

Figure 2: Evolution of neural network weights w_t and tangent linear model posterior mean $v^\star$ during training. With online Laplace (OL procedure), w_t and $v^\star$ converge to the same distribution. Offline training does not exhibit this behaviour.

Online hyperparameter optimisation with the tangent linear model The mode-corrected evidence approximation $\mathcal{L}_h(\mathcal{M}; w_t)$ matches the exact evidence of a *tangent model* with regressor $h : \mathbb{R}^{d_w} \times \mathbb{R}^{d_x} \to \mathbb{R}$ given by a first-order Taylor approximation to f at w_t

$$y = h(v) + \epsilon \;\text{ with }\; h(v) := f(w_t) + J(v - w_t) \;\text{ and }\; v \sim \mathcal{N}(0, \alpha^{-1} I_{d_w}), \;\; \epsilon \sim \mathcal{N}(0, \beta^{-1} I_n).$$

The tangent model's loss is $\ell_h(v; \mathcal{M}) = \frac{\beta}{2}\|y - h(v)\|^2 + \frac{\alpha}{2}\|v\|^2$. It is minimised by the MAP $v^\star := \arg\min_v \ell_h(v; \mathcal{M})$ and its constant curvature matches the GGN $\partial_v^2 \ell_h(v^\star; \mathcal{M}) = H$.

The mode-corrected algorithm selects hyperparameters that improve the tangent model's evidence at each step. We hereon refer to this procedure as online Linear Model (LM). Since the NN and linear model share hyperparameters $\mathcal{M}$, their loss gradients match at the linearisation point $\partial_w \ell_f(w_t; \mathcal{M}) = \partial_v \ell_h(w_t, \mathcal{M})$. Thus, the NN optimisation step brings its parameters closer to the MAP of the current tangent model while also changing the linearisation point, which in turn changes the tangent model. This is not guaranteed to result in a tangent linear model with increased evidence. Indeed, Figure 3 shows the LM evidence exhibits non-monotonic training curves.

The LM procedure reaches a fixed point when a simultaneous optimum of the NN weights and linear model hyperparameters is found: $w^\star = \arg\min_w \ell_f(w; \mathcal{M}^\star)$ and $\mathcal{M}^\star = \arg\max_{\mathcal{M}} \mathcal{L}_h(\mathcal{M}; w^\star)$. Since the first-order term vanishes here, $\partial_w \ell_f(w^\star, \mathcal{M}) = 0$, the linear model evidence $\mathcal{L}_h$ matches the regular Laplace evidence $\mathcal{L}_f$, satisfying the termination condition of Mackay (1992)'s NN retraining algorithm. Additionally, since the tangent model is expanded around $w^\star$, we have $\partial_w \ell_f(w^\star, \mathcal{M}) = \partial_v \ell_h(w^\star, \mathcal{M}) = 0$. We may use similarity between $w^\star$ and $v^\star$ to check for convergence. Figure 2 shows the LM procedure results in the NN and tangent model converging to a shared MAP. This convergence does not occur when performing standard NN training with fixed hyperparameters.

Online Laplace as a variational bound on the tangent model's evidence Finally, we connect the LM procedure to the OL procedure described in Section 2.1. For this, we

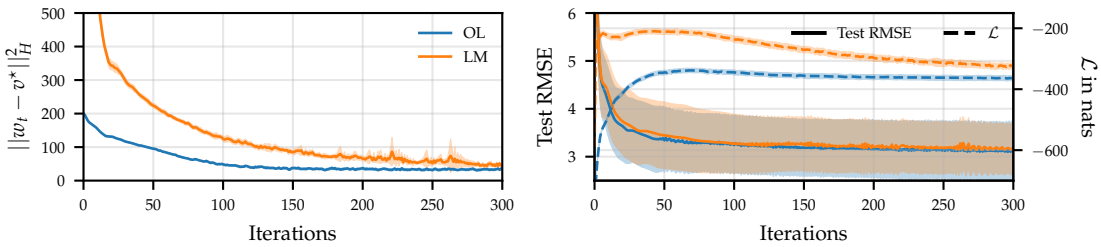

Figure 3: Difference between neural network weights w_t and linear model posterior mean $v^\star$ (left) and test RMSE versus ELBO $\mathcal{L}$ (right) throughout online training for both OL and LM procedures. A maximised ELBO does not imply optimal test RMSE.

construct a lower bound (ELBO) on the linear model's evidence,

$$\mathcal{L}_h(\mathcal{M}; w_t) \geq \mathbb{E}_{\mathcal{N}(v;\,\mu, H^{-1})}[\log p_h(y, v; \mathcal{M})] + \mathcal{H}(\mathcal{N}(\mu, H^{-1})) := \mathcal{L}(\mathcal{M}; \mu, w_t) \tag{3}$$
$$= \exp\left(\frac{d_w}{2}\log\alpha + \frac{n}{2}\log\beta - \frac{\beta}{2}\|y - (f(w_t) + J(\mu - w_t))\|_2^2 - \frac{\alpha}{2}\|\mu\|_2^2 - \frac{1}{2}\log\det H - \frac{n}{2}\log 2\pi\right)$$

where $\mathcal{H}$ refers to the differential entropy. The bound is tight when the variational posterior mean μ matches the linear model's MAP $v^\star$, that is, $\mathcal{L}(\mathcal{M}; v^\star, w_t) = \mathcal{L}_h(\mathcal{M}; w_t)$.

By choosing the variational mean to match the linearisation point, $\mu = w_t$, we recover the OL evidence $\mathcal{L}(\mathcal{M}; w_t, w_t) = \mathcal{L}_f(\mathcal{M}; w_t)$. Thus, discarding the first-order term in the Taylor expansion of $\log p_f(y, w; \mathcal{M})$, which seemed unjustified, actually results in a lower bound on a variant of the Laplace evidence which corrects for the expansion point not matching the mode, that is, $\mathcal{L}_f(\mathcal{M}; w_t) \leq \mathcal{L}_h(\mathcal{M}; w_t)$. This bound is illustrated in Figure 4 (left).

The OL procedure reaches a fixed point at the simultaneous optimum $w^\star = \arg\min_w \ell_f(w; \mathcal{M}^\star)$ and $\mathcal{M}^\star = \arg\max_{\mathcal{M}} \mathcal{L}(\mathcal{M}; w^\star, w^\star)$. Here, both the ELBO is tight and the first-order term vanishes. Thus OL shares a fixed point with the LM procedure and Mackay (1992)'s iterative NN retraining algorithm. During training, the slack in the OL bound results in smaller hyperparameter steps, as shown in Figure 4, leading to more stable convergence to a shared MAP for the NN and tangent model than the LM procedure.

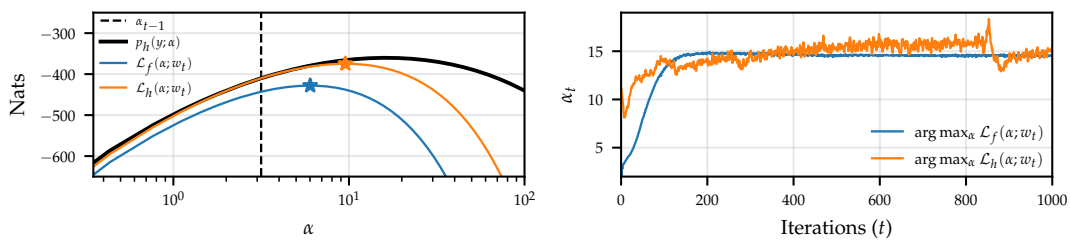

Figure 4: Illustration of exact linear model evidence, and LM and OL hyperparameter objectives at a single train step t (left). $\mathcal{L}_h$ is a tight bound but leading to larger updates which make hyperparameter trajectories unstable (right).

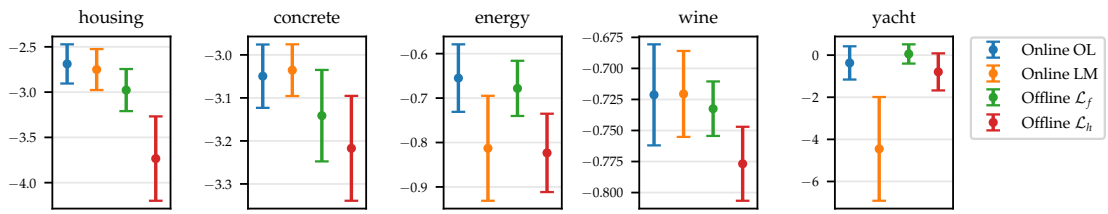

Figure 5: Test log-likelihood on UCI regression (mean ± standard error over 10 splits).

4. Experiments

This section demonstrates the OL and LM online hyperparameter tuning algorithms discussed in the previous section and contrasts them with standard NN training, that is, with fixed hyperparameters (labelled offline). To this end, we train single hidden layer, 50 hidden unit MLPs on 5 UCI datasets (Dua and Graff, 2017): housing, concrete, energy, wine, and yacht. We train our NNs with full-batch Adam and an exponentially decaying learning rate. For online methods, we train until convergence, updating the hyperparameters after every NN parameter step. The hyperparameter update step is detailed in Appendix C. For offline methods, we select the stopping point that minimises RMSE on a validation set consisting of 10% of training data points. We do not use normalisation layers, as these interfere with the Laplace approximation (Antorán et al., 2022a).

Convergence of NN and linear model posteriors Figure 3 (left) shows the Euclidean distance between the NN parameters and tangent model MAP decreasing throughout training for both the LM and OL procedures. Although the distance does not exactly reach 0, the NN and linear model parameter vectors present nearly identical distributions, as shown in Figure 2. This does not occur in the offline setting. The LM procedure takes larger hyperparameter steps than OL resulting in some training instability and overall slower convergence, as shown in Figure 4. Figure 3 (right) shows that both the LM and OL evidence estimates attain their maxima very early in training, before the lowest test RMSE is reached. As discussed in Section 3, these metrics target the evidence of a tangent model and may only transfer to the NN once online training has fully converged.

Predictive performance We obtain predictive distributions using the linearised Laplace method (Immer et al., 2021b). For NNs trained offline, linearised Laplace hyperparameters are chosen post-hoc my maximising either the $\mathcal{L}_f$ or $\mathcal{L}_h$ objectives. Figure 5 reports log-likelihood mean and standard errors obtained across 10 train-test splits where the test set size is 10% of the whole dataset size.

We find the OL procedure to consistently provide the best performance or be a close second best. The LM procedure performs very similarly to OL on all datasets except energy and yacht, where it performs the worse out of all methods. We attribute this to training instability. Offline methods only perform competitively on yacht, despite use of validation-based stopping. The online methods' improvement over their offline counterparts can mostly be explained by improved RMSE (see Figure 14), suggesting online hyperparameter optimisation effectively navigates the bias-variance trade off.

The full set of experimental results, including hyperparameter trajectories and evidence estimates is provided in Appendix A.

References

Javier Antorán, James Urquhart Allingham, David Janz, Erik Daxberger, Eric Nalisnick, and José Miguel Hernández-Lobato. Linearised laplace inference in networks with normalisation layers and the neural g-prior. In *Fourth Symposium on Advances in Approximate Bayesian Inference*, 2022a. URL https://openreview.net/forum?id=uUH8x-h9zdB.

Javier Antorán, David Janz, James Urquhart Allingham, Erik A. Daxberger, Riccardo Barbano, Eric T. Nalisnick, and José Miguel Hernández-Lobato. Adapting the linearised laplace model evidence for modern deep learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 796–821. PMLR, 2022b. URL https://proceedings.mlr.press/v162/antoran22a.html.

Javier Antoran, Shreyas Padhy, Riccardo Barbano, Eric Nalisnick, David Janz, and José Miguel Hernández-Lobato. Sampling-based inference for large linear models, with application to linearised laplace. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=aoDyX6vSqsd.

Riccardo Barbano, Javier Antorán, Johannes Leuschner, José Miguel Hernández-Lobato, Zeljko Kereta, and Bangti Jin. Fast and painless image reconstruction in deep image prior subspaces. *CoRR*, abs/2302.10279, 2023. doi: 10.48550/arXiv.2302.10279. URL https://doi.org/10.48550/arXiv.2302.10279.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 557–565. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/botev17a.html.

Ross M Clarke, Elre Talea Oldewage, and José Miguel Hernández-Lobato. Scalable one-pass optimisation of high-dimensional weight-update hyperparameters by implicit differentiation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hfU7Ka5cfrC.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

F. Dan Foresee and Martin T. Hagan. Gauss-newton approximation to bayesian learning. In *Proceedings of International Conference on Neural Networks (ICNN'97), Houston, TX, USA, June 9-12, 1997*, pages 1930–1935. IEEE, 1997. doi: 10.1109/ICNN.1997.614194. URL https://doi.org/10.1109/ICNN.1997.614194.

Karl J. Friston, Jérémie Mattout, Nelson J. Trujillo-Barreto, John Ashburner, and William D. Penny. Variational free energy and the laplace approximation. *NeuroImage*, 34(1):220–234, 2007. doi: 10.1016/j.neuroimage.2006.08.035. URL https://doi.org/10.1016/j.neuroimage.2006.08.035.

Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4563–4573. PMLR, 2021a. URL http://proceedings.mlr.press/v139/immer21a.html.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 703–711. PMLR, 2021b. URL http://proceedings.mlr.press/v130/immer21a.html.

Alexander Immer, Tycho F.A. van der Ouderaa, Gunnar Ratsch, Vincent Fortuin, and Mark van der Wilk. Invariance learning in deep neural networks with differentiable laplace approximations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=i3ewAfTbCxJ.

David John Cameron Mackay. *Bayesian Methods for Adaptive Models*. PhD thesis, USA, 1992.

James Martens. New insights and perspectives on the natural gradient method. *J. Mach. Learn. Res.*, 21:146:1–146:76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

Appendix A. Evolution of neural network training

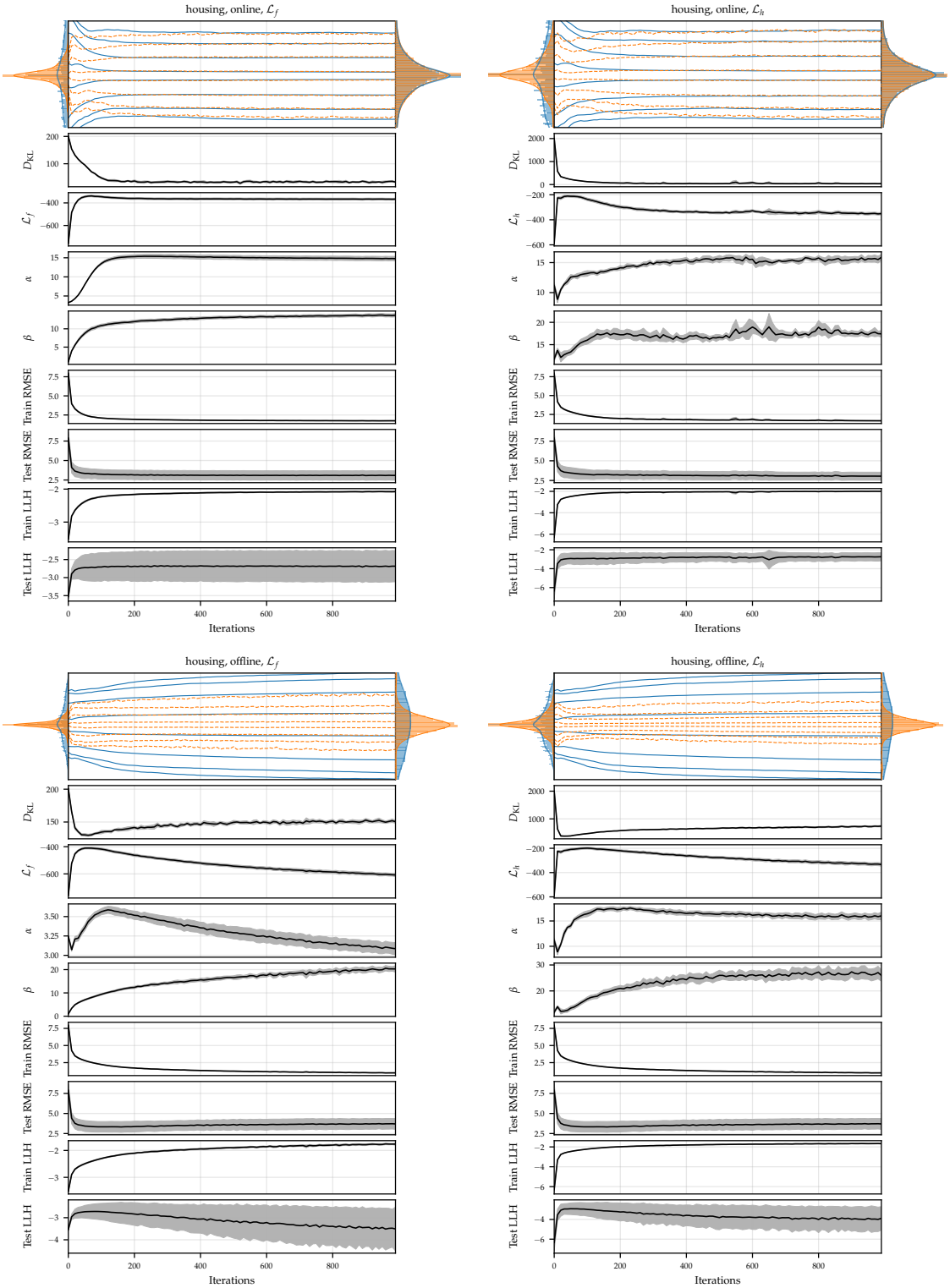

Figure 6: Evolution of neural network training on the housing dataset.

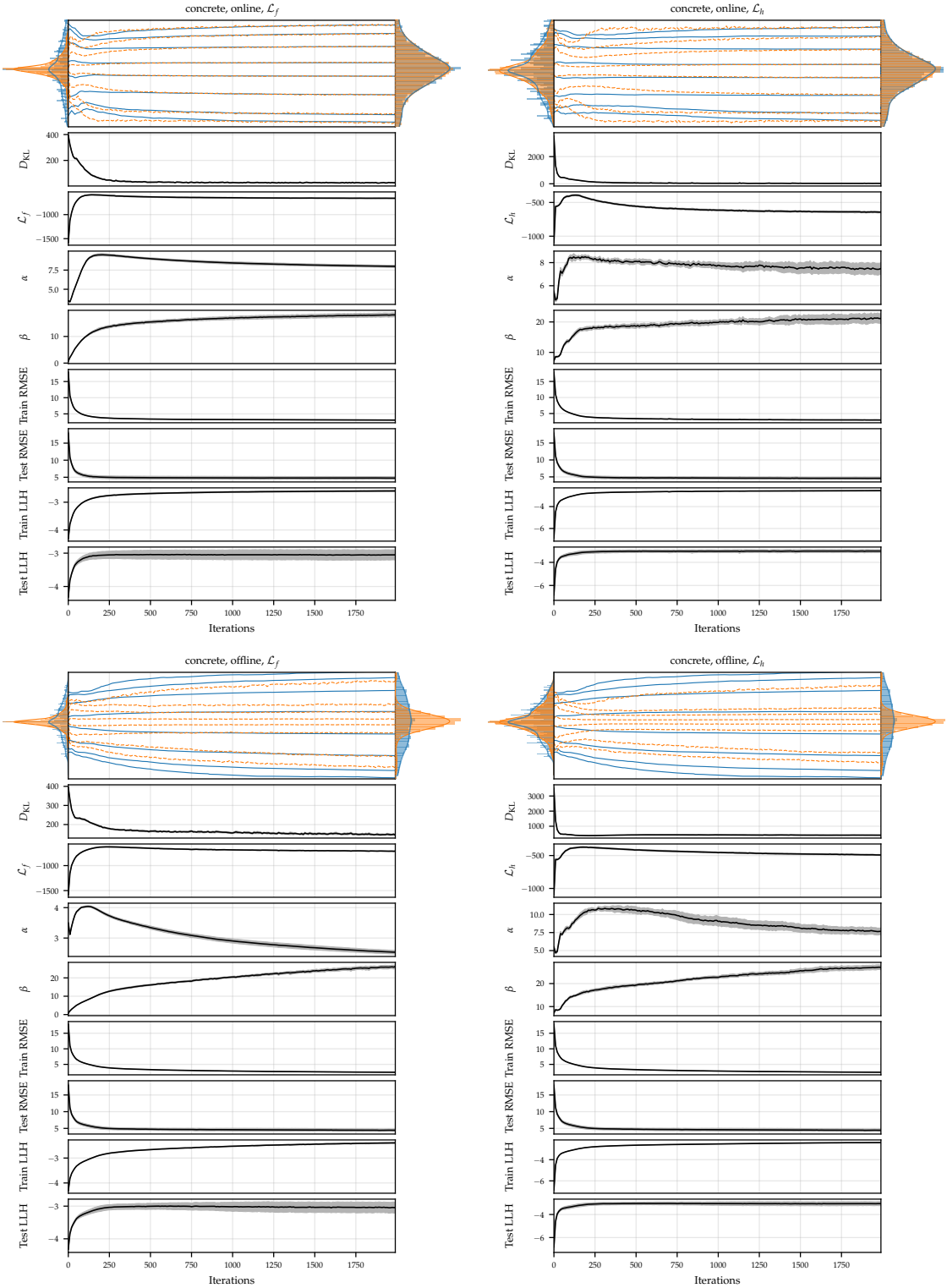

Figure 7: Evolution of neural network training on the concrete dataset.

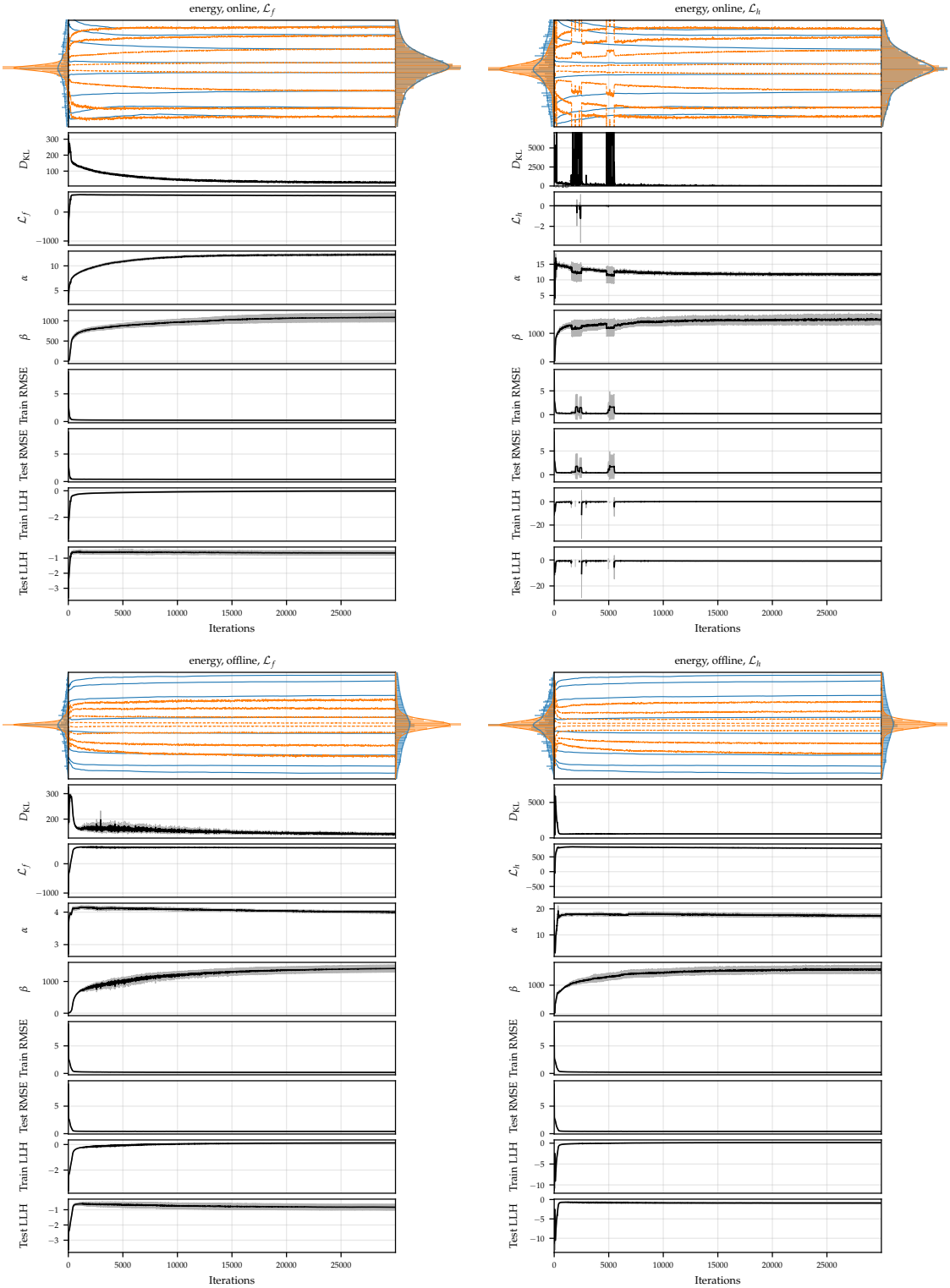

Figure 8: Evolution of neural network training on the energy dataset.

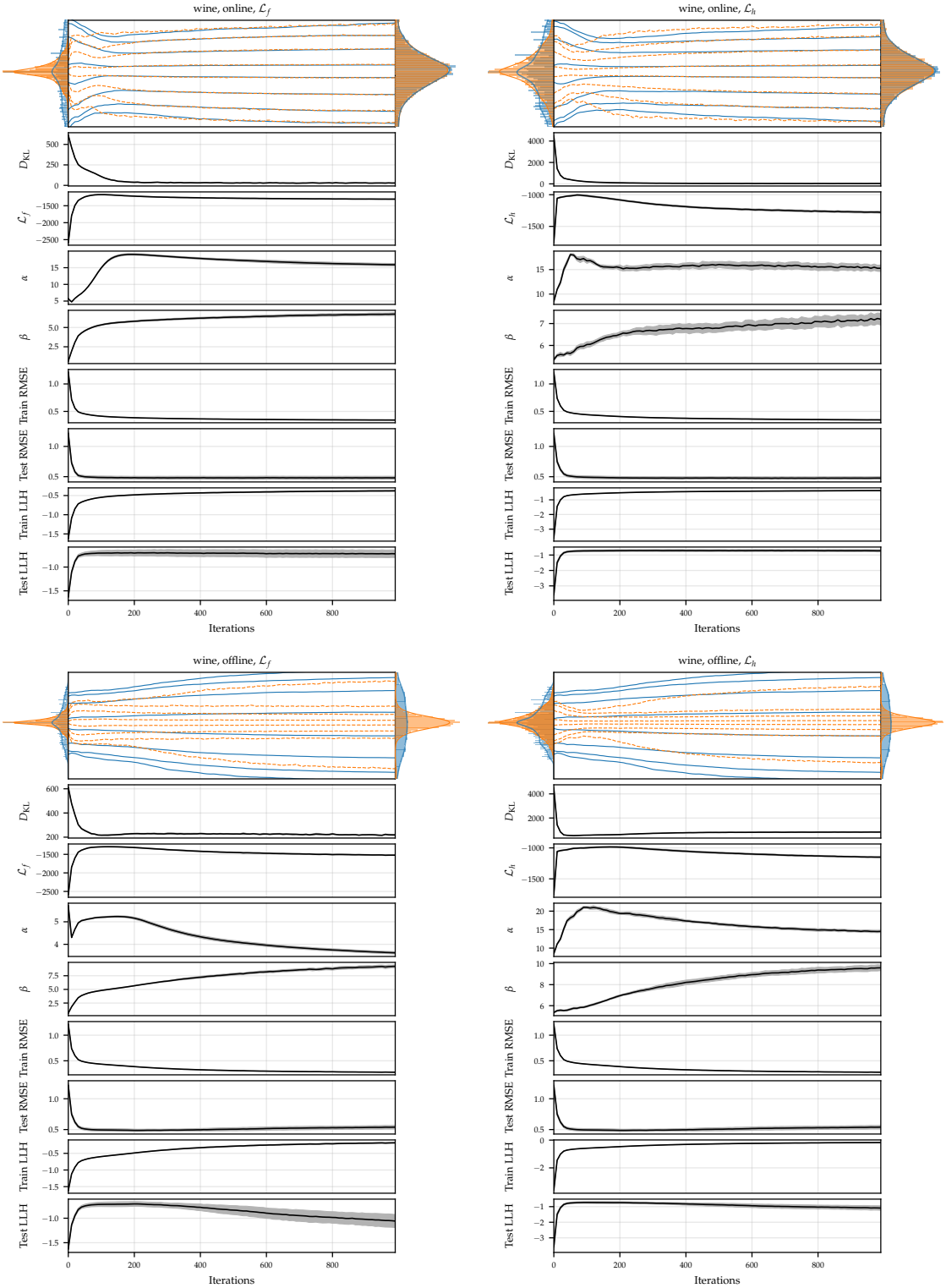

Figure 9: Evolution of neural network training on the wine dataset.

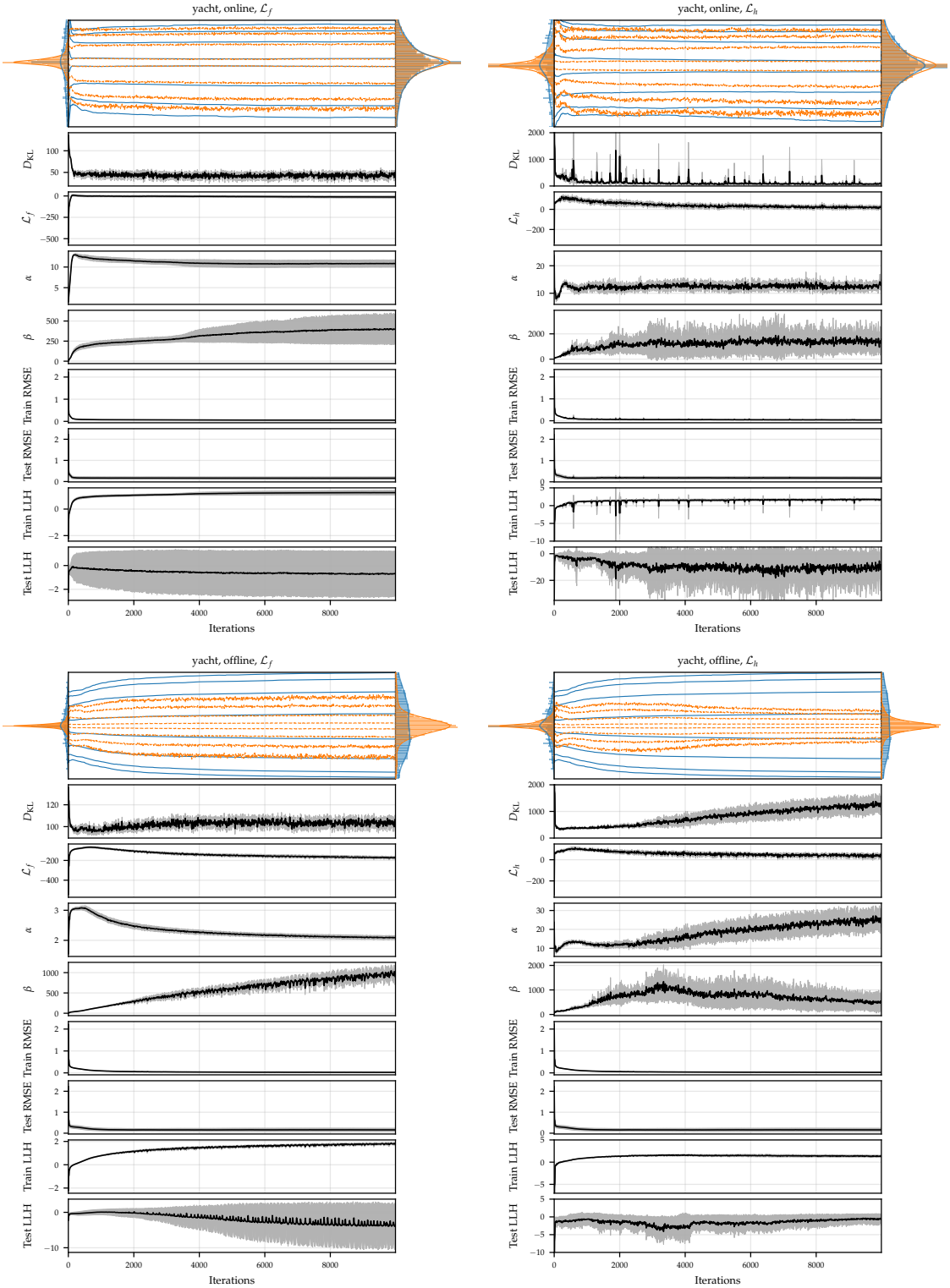

Figure 10: Evolution of neural network training on the yacht dataset.

Appendix B. Additional predictive metrics

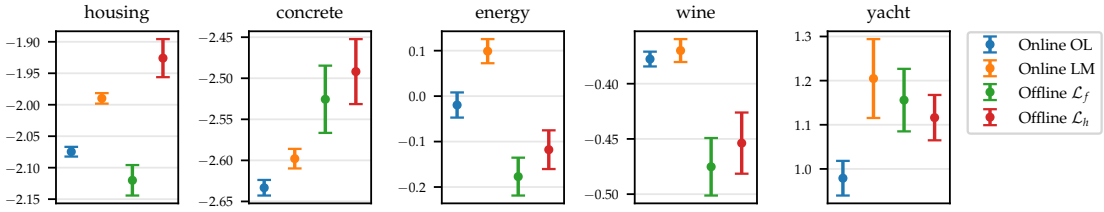

Figure 11: Train log-likelihood on UCI regression (mean ± standard error over 10 splits).

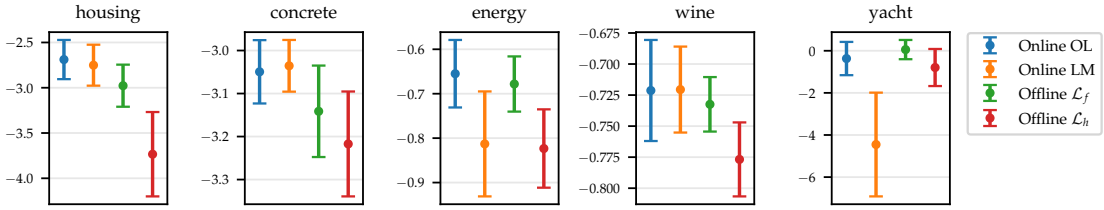

Figure 12: Test log-likelihood on UCI regression (mean ± standard error over 10 splits).

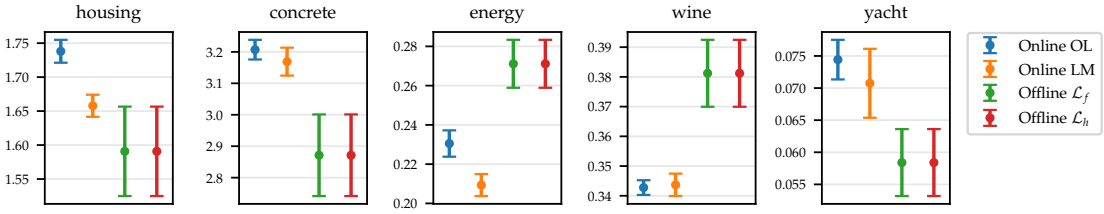

Figure 13: Train RMSE on UCI regression (mean ± standard error over 10 splits).

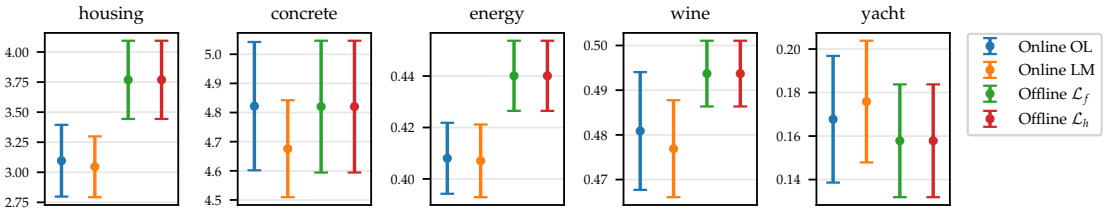

Figure 14: Test RMSE on UCI regression (mean ± standard error over 10 splits).

Appendix C. MacKay's hyperparameter update

Given mean μ and predictions $\hat{y}$, the hyperparameters α and β can be updated by

$$\alpha = \frac{\gamma}{\|\mu\|_2^2}, \qquad \beta = \frac{n-\gamma}{\|y-\hat{y}\|_2^2}, \qquad \gamma = d_w - \alpha \mathrm{Tr}(H^{-1}), \tag{4}$$

to maximise the linear model's evidence (Mackay, 1992). For OL, $\mu = w$ and $\hat{y} = f(w)$, and in the case of LM, $\mu = v^\star$ and $\hat{y} = h(v^\star)$.

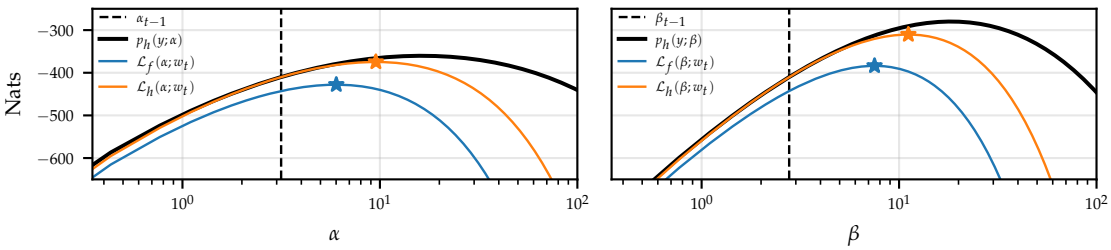

Figure 15: Illustration of a hyperparameter update at a single timestep t. Starting at the current value (dashed black), the updated value (star) is obtained by maximising the ELBO.

Appendix D. Implementation Details

The initial learning rate used for Adam was set to 0.01 and the exponential learning rate decay factor was set to 0.9999. In the online setting, housing and wine were trained for 1000, concrete for 2000, energy for 30000 and yacht for 10000 steps until convergence. In the offline setting, the validation RMSE was tracked for a maximum of 30000 epochs.