Repulsive Ensembles for Bayesian Inference in Physics-Informed Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Physics-informed neural networks (PINNs) have proven an effective tool for solving differential equations, in particular when considering non-standard or ill-posed settings. When inferring solutions and parameters of the differential equation from data, uncertainty estimates are preferable to point estimates, as they give an idea about the accuracy of the solution. In this work, we consider the inverse problem and employ repulsive ensembles of PINNs (RE-PINN) for obtaining such estimates. The repulsion is implemented by adding a particular repulsive term to the loss function, which has the property that the ensemble predictions correspond to the true Bayesian posterior in the limit of infinite ensemble members. Where possible, we compare the ensemble predictions to Monte Carlo baselines. Whereas the standard ensemble tends to collapse to maximum-a-posteriori solutions, the repulsive ensemble produces significantly more accurate uncertainty estimates and exhibits higher sample diversity.

1 Introduction

Physics-informed neural networks were introduced by (Raissi et al., 2019) and have become a thriving area of research since then (Cuomo et al., 2022; Cai et al., 2021a; Lawal et al., 2022; Cai et al., 2021b). They constitute an alternate approach to solving differential equations (DEs): a neural network is trained to learn the solution, where the DE is included as a soft constraint during network training. In practical applications, it is of particular importance to obtain uncertainty estimates for the model predictions. This is typically addressed in a Bayesian framework (Box and Tiao, 2011) and has led to the development of Bayesian neural networks (BNNs) (Neal, 2012; Jospin et al., 2022), and thereafter Bayesian PINNs (B-PINNs) (Yang et al., 2021). Since exact inference is intractable, alternative methods such as Markov Chain Monte Carlo (MCMC) (Neal et al., 2011), variational inference (VI) (Graves, 2011), and dropout (Gal and Ghahramani, 2016) are often used.

Deep ensembles (Lakshminarayanan et al., 2017) have emerged as a promising alternative to these approaches for uncertainty estimation. Multiple networks are trained in parallel, and, due to randomness in initialization and training, they can end up in different local optima. Assuming standard training losses, these solutions correspond to maximum-a-posteriori (MAP) estimates of the posterior. To encourage diversity between the ensemble members, adding a repulsive term to the loss function that prevents them from collapsing

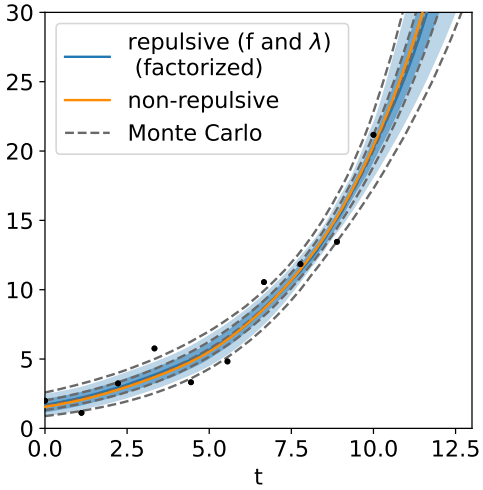

Figure 1: Distribution of the repulsive ensemble when solving the exponential equation. The ensemble median is depicted, together with the 0.1-0.9 quantiles, and the 0.25-0.75 quantiles (shaded areas). The non-repulsive ensemble collapses to the MAP estimate. The dashed lines correspond to the MC baseline. The black dots depict training data. The ensembles have 50 members.

to the same solution has proven fruitful (Liu and Wang, 2016). Due to the non-identifiability of neural networks (Roeder et al., 2021), i.e., the fact that different configurations of the network parameters (consider, e.g., permutations of hidden units) can lead to the same predictions in function space, it is especially effective to add this repulsion directly in function space (Wang et al., 2019), rather than in weight space. D'Angelo and Fortuin (2021) then gave justification for repulsive ensembles from the Bayesian point of view. Building on connections between particle methods and deep ensembles (Jordan et al., 1998; Wang et al., 2019), they derived a specific form for the repulsive term that guarantees convergence of the ensemble distribution to the true Bayesian posterior in the limit of infinite ensemble members.

In this work, we develop a method for uncertainty estimation in PINNs that corresponds to the Bayesian posterior. That is, the method quantifies the uncertainty in the predictions that arise due to finite and noisy data. We consider the inverse problem, where the PINN learns the DE solution jointly with the DE parameters from data. Deep ensembles constitute a promising candidate for this task: since each ensemble member is individually constrained to fulfill the DE with its own set of DE parameters, each will constitute a valid DE solution. In turn, information about the DE solution space will be captured in the uncertainty intervals. This would, in general, not be the case when using VI, which requires the posterior to belong to a predefined family of variational distributions, nor when using dropout techniques to estimate uncertainty, which have been shown to be detrimental to PINN performance (Yang et al., 2021). In this work, we apply the approach of D'Angelo and Fortuin (2021) to PINNs and investigate how its properties are affected by the additional DE loss term and parameters. We show that repulsion needs to be applied in the joint space of function values and DE parameters, according to theory. We conduct experiments on different datasets and evaluate the effectiveness of different approximations of the repulsive term.

In Figure 1, the performance of our approach is demonstrated for the exponential differential equation. It is apparent that the repulsive ensemble of PINNs (RE-PINN) agrees very well with the Monte Carlo (MC) baseline, both in terms of the median prediction as well as the uncertainty intervals, whereas the non-repulsive ensemble essentially collapses to the MAP estimate.

2 Related work

Yang et al. (2021) introduce a Bayesian formulation of PINNs (B-PINN) that builds on Bayesian NNs (Neal, 2012; Jospin et al., 2022), and where PDE residuals are treated as low-noise measurements. They show that Hamiltonian Monte Carlo gives good uncertainty estimates, whereas variational inference (VI) with mean-field Gaussian approximation, and dropout lead to unreasonable estimates. In Li et al. (2024), they apply B-PINN with VI to seismic data.

In Jiang et al. (2023), ensembles of PINNs (E-PINNs) are considered for solving inverse heat conduction problems. They combine the PINN training with an adaptive sampling scheme to improve the predictive capabilities of the model. In Sahli Costabal et al. (2020), ensembles of PINNs with randomized priors are used for active learning in cardiac activation mapping. The uncertainty estimates obtained from the ensemble are utilized to suggest new measurements.

In Haitsiukevich and Ilin (2023), ensembles of PINNs are used as a means to improve the quality of the PINN solution. The agreement between the ensemble members is used as a metric to ensure accurate solutions before expanding the domain of the solution. In Zou et al. (2025), they propose ensembles of PINNs as a means to solve ODEs and PDEs with solution multiplicity, since ensemble members with different initializations can converge to different solutions. In Fang et al. (2023), they develop a technique inspired by classical gradient-boosting techniques; the members of a PINN ensemble are trained sequentially to improve upon the previous predictions.

In Yang and Foster (2022), confidence intervals for PINNs are obtained by having the PINN predict multiple (hundreds of) outputs, which are supposed to reflect the uncertainty in the prediction. In Tan et al. (2025), the principles of evidential deep learning are employed to obtain uncertainty estimates.

In Röver et al. (2024), repulsive ensembles of PINNs are used to estimate the Hubble function from supernova data. While they also employ repulsive ensembles to obtain uncertainty estimates, they do not infer DE

Method	MC	VI	ensemble	repulsive ensemble	other	Reference
B-PINN	✓	✓				Yang et al. (2021)
BPINN-VI		✓				Li et al. (2024)
MO-PINN	✓				✓	Yang and Foster (2022)
E-PINN			✓			Jiang et al. (2023)
Multi-solution PINN			✓			Zou et al. (2025)
EikonalNet			✓			Sahli Costabal et al. (2020)
Functional repulsion PINN			✓	✓		Röver et al. (2024)
RE-PINN	✓	✓	✓	✓		our paper

Table 1: Overview of works that consider uncertainty quantification in PINNs. Green ticks indicate methods that constitute integral parts of the method described in the paper. Grey ticks indicate methods that were considered for comparison.

parameters and only apply repulsion in function space. The different works discussed in this section and the methods of uncertainty quantification considered in them are summarized in Table 1.

3 Background: PINNs in the Bayesian Setting

In this paper, we employ PINNs (Raissi et al., 2019) to solve differential equations of the form

$$\mathcal{F}(f, \lambda)(x) = 0, \tag{1}$$

where $\mathcal{F}$ is the differential operator defining the differential equation with parameters λ, and where $f(x)$ is the solution of the differential equation at inputs x, which often contain time or spatial coordinates. For example, the differential operator corresponding to the exponential equation from Figure 1 is given by $\mathcal{F}(f, \lambda)(\cdot) = \frac{\partial f(\cdot)}{\partial x} - \lambda f(\cdot)$. Both f and λ are to be inferred from data. The solution is modeled as a neural network $\hat{f}(x) := \hat{f}(x; \theta)$ with network parameters θ. Model predictions for the DE parameters are denoted by $\hat{\lambda}$.

We follow the approach of Yang et al. (2021) to describe PINNs in the Bayesian setting. The data is given by $\mathcal{D}_d = \{(x_j, y_j)\}_{j=1}^{N_d}$, where the y_j denote noisy measurements of $f(x_j)$. In addition, we choose another dataset $\mathcal{D}_c = \{(x_j, 0)\}_{j=1}^{N_c}$, which contains collocation points at which fulfillment of the DE is enforced during training. The zeros in this dataset are to be interpreted as noisy measurements of the right-hand side of (1) (i.e., we assume that we may be mistaken about the source term being exactly zero), with standard deviation $\sigma_{\mathcal{F}}$. The two datasets are then collected in $\mathcal{D} = \{\mathcal{D}_d, \mathcal{D}_c\}$.

Since we will eventually consider repulsion in function space, we directly consider the formulation in function space. The posterior distribution $p(f, \lambda|\mathcal{D})$ is employed as a means to systematically determine the loss function. With Bayes' law,

$$p(f, \lambda|\mathcal{D}) = \frac{p(\mathcal{D}|f, \lambda)p(f, \lambda)}{p(\mathcal{D})}, \tag{2}$$

and taking the negative logarithm, we obtain

$$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_{\mathcal{F}} - \log p(f, \lambda) + \log p(\mathcal{D}), \tag{3}$$

where $\mathcal{L}_f$ and $\mathcal{L}_{\mathcal{F}}$ denote the likelihoods for the different data modalities at hand. Assuming Gaussian noise and omitting constants, they are given by

$$\mathcal{L}_f = \frac{1}{2\sigma_f^2} \sum_{j=1}^{N_d} (\hat{f}(x_j) - y_j)^2, \tag{4a}$$

$$\mathcal{L}_{\mathcal{F}} = \frac{1}{2\sigma_{\mathcal{F}}^2} \sum_{j=1}^{N_c} (\mathcal{F}(\hat{f}, \hat{\lambda})(x_j))^2. \tag{4b}$$

In case of different noise types, (4a) would need to be replaced with the corresponding negative log-likelihood. The loss (4b) is equivalent to the soft loss term from the original PINN formulation (Raissi et al., 2019).

4 Method

The aim of our method is to obtain an estimate of the Bayesian posterior (2) for the function values f and DE parameters λ. When training an ensemble of neural networks with loss function (3), this will not yield samples from the Bayesian posterior, but instead a collection of MAP estimates (Lakshminarayanan et al., 2017).

D'Angelo and Fortuin (2021) addressed this issue for standard neural networks. By considering the Wasserstein gradient flow (Ambrosio et al., 2008) when minimizing the KL-divergence between the ensemble distribution $\rho(\boldsymbol{f})$, and the true posterior $p(\boldsymbol{f}|\mathcal{D})$, they derived an expression for the function value updates of the i-th ensemble member (see Appendix A.1 for details):

$$\boldsymbol{f}_i^{t+1} = \boldsymbol{f}_i^t + \epsilon_t \left(\nabla_f \log p(\boldsymbol{f}_i^t|\mathcal{D}) - \nabla_f \log \rho(\boldsymbol{f}_i^t)\right). \tag{5}$$

This expression now includes a term involving the density of the current ensemble predictions, which acts as a repulsive force between the ensemble members. The vector $\boldsymbol{f}_i^t$ contains the function values $\hat{f}_i(x_j)$ of the i-th ensemble member at the data points $x_j \in \mathcal{D}_d$. This discretization is necessary since continuous functions are intractable.

When considering the inverse problem in PINNs, posterior estimates for λ as well as f are needed. Due to the qualitative differences between the network parameters θ, which result in the function values $\boldsymbol{f}$, and the DE parameters λ, which only show up in the loss (4b), the modification of (5) requires careful consideration. The key point is that $\boldsymbol{f}$ and λ do not explicitly depend on each other in PINNs. Since the derivation of (5) does not make restrictive assumptions on the quantities involved, this means that an analogous expression to (5) can be obtained by substituting $[\boldsymbol{f}^\mathsf{T}, \lambda^\mathsf{T}]^\mathsf{T}$ for $\boldsymbol{f}$:

$$\begin{bmatrix} \boldsymbol{f}_i^{t+1} \\ \lambda_i^{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{t} \\ \lambda_i^{t} \end{bmatrix} + \epsilon_t \begin{bmatrix} \nabla_f (\log p(\boldsymbol{f}_i^t, \lambda_i^t|\mathcal{D}) - \log \rho(\boldsymbol{f}_i^t, \lambda_i^t)) \\ \nabla_\lambda (\log p(\boldsymbol{f}_i^t, \lambda_i^t|\mathcal{D}) - \log \rho(\boldsymbol{f}_i^t, \lambda_i^t)) \end{bmatrix}. \tag{6}$$

Making use of (2) and plugging in the expressions from (4), we obtain (after omitting constants)

$$\begin{bmatrix} \boldsymbol{f}_i^{t+1} \\ \lambda_i^{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{t} \\ \lambda_i^{t} \end{bmatrix} + \epsilon_t \begin{bmatrix} \nabla_f \big(-\mathcal{L}_{fi} - \mathcal{L}_{\mathcal{F}i} + \log p(\boldsymbol{f}_i^t, \lambda_i^t) - \log \rho(\boldsymbol{f}_i^t, \lambda_i^t)\big) \\ \nabla_\lambda \big(-\mathcal{L}_{\mathcal{F}i} + \log p(\boldsymbol{f}_i^t, \lambda_i^t) - \log \rho(\boldsymbol{f}_i^t, \lambda_i^t)\big) \end{bmatrix}. \tag{7}$$

Since neural networks are not trained directly in function space, an expression for the corresponding updates in weight space is required. This can be done by projecting the equation to the network weights θ:

$$\begin{bmatrix} \theta_i^{t+1} \\ \lambda_i^{t+1} \end{bmatrix} = \begin{bmatrix} \theta_i^{t} \\ \lambda_i^{t} \end{bmatrix} + \epsilon_t \begin{bmatrix} \left(\frac{\partial \boldsymbol{f}_i^t}{\partial \theta_i^t}\right)^\mathsf{T} \nabla_f \big(-\mathcal{L}_{fi} - \mathcal{L}_{\mathcal{F}i} + \log p(\boldsymbol{f}_i^t, \lambda_i^t) - \log \rho(\boldsymbol{f}_i^t, \lambda_i^t)\big) \\ \nabla_\lambda \big(-\mathcal{L}_{\mathcal{F}i} + \log p(\boldsymbol{f}_i^t, \lambda_i^t) - \log \rho(\boldsymbol{f}_i^t, \lambda_i^t)\big) \end{bmatrix}, \tag{8}$$

where $\frac{\partial \boldsymbol{f}_i^t}{\partial \theta_i^t}$ denotes the Jacobian. Since neither λ enters directly into $\boldsymbol{f}$, nor vice versa, no projection is required for the λ updates, and no λ-derivatives are required when calculating the Jacobian of $\boldsymbol{f}$. Both quantities interact only indirectly, through the likelihood term $\mathcal{L}_{\mathcal{F}}$, as well as the prior and the repulsive term.

When training our ensemble of PINNs, (8) implies the following loss function for the i-th ensemble member:

$$\mathcal{L}_i = \mathcal{L}_{fi} + \mathcal{L}_{\mathcal{F}i} - \log p(\boldsymbol{f}_i, \lambda_i) + \log \rho(\boldsymbol{f}_i, \lambda_i). \tag{9}$$

The Jacobian from (8) does not show up explicitly in the loss, since it will automatically be taken into account through the chain rule. After normalizing the loss with respect to the number of data points N_d and rescaling, we obtain

$$\begin{aligned} \tilde{\mathcal{L}}_i = &\frac{1}{N_d} \sum_{j=1}^{N_d} \left(\hat{f}_i(x_j) - y_j\right)^2 + \frac{f_\lambda}{N_c} \sum_{j=1}^{N_c} \left(\mathcal{F}(\hat{f}_i(x_j), \lambda_i)\right)^2 \\ &+ f_\rho \big(-\log p(\boldsymbol{f}_i, \lambda_i) + \log \rho(\boldsymbol{f}_i, \lambda_i)\big), \end{aligned} \tag{10}$$

Repulsion	repulsive term
non-repulsive	$\log \rho(\boldsymbol{f}, \lambda) \to 0$
repulsive (f)	$\log \rho(\boldsymbol{f}, \lambda) \to \log \rho(\boldsymbol{f})$
repulsive (λ)	$\log \rho(\boldsymbol{f}, \lambda) \to \log \rho(\lambda)$
repulsive (f, λ)	$\log \rho(\boldsymbol{f}, \lambda)$
repulsive (f, λ) - factorized	$\log \rho(\boldsymbol{f}, \lambda) \approx \log \rho_f(\boldsymbol{f}) + \log \rho_\lambda(\lambda)$
repulsive (f, λ) - fully factorized	$\log \rho(\boldsymbol{f}, \lambda) \approx \frac{1}{\sqrt{N_d}} \left(\sum_j \log \rho_{fj}(f_j) + \sum_l \log \rho_{\lambda l}(\lambda_l) \right)$

Table 2: Different variants of repulsive ensembles that we considered in our experiments. The first three models denote variants that we implement for the sake of comparison, where repulsion is either omitted or only takes place in a subspace. The last three models correspond to different ways of approximating the full repulsive term.

where $f_\lambda = \frac{\sigma_f^2 N_c}{\sigma_\mathcal{F}^2 N_d}$ and $f_\rho = \frac{2\sigma_f^2}{N_d}$.

Evaluating the loss (10) requires knowledge of the density ρ, which is unavailable. Hence, we utilize a particle approximation, that is, we approximate the distribution via the ensemble members. For this, we have used kernel density approximation (KDE, see Appendix B). As pointed out in D'Angelo and Fortuin (2021), KDE becomes exact in the limit of infinite ensemble members; this implies that minimizing the loss (10) would lead to the true Bayesian posterior, also in our case.

The prior $p(\boldsymbol{f}, \lambda)$ can be chosen freely and in a way that the evaluation will be straightforward. The approximation of the density $\rho(\boldsymbol{f}, \lambda)$ via KDE, however, needs special consideration. Since $\boldsymbol{f}$ and λ are rather different in nature, this may affect the performance of a naive approximation via KDE. One way of improving the approximation could be to approximate the density as $\rho(\boldsymbol{f}, \lambda) \approx \rho_f(\boldsymbol{f})\rho_\lambda(\lambda)$; then, KDE could be performed separately for $\boldsymbol{f}$ and λ. Furthermore, the dimensionality of $\boldsymbol{f}$ may be large, which can also lead to bad KDE estimates. Factorizing the density further between different function values could improve the estimates.

In Table 2, we summarize the various approximations and modifications for the density term that we consider in our experiments. We also include variants where the repulsion happens in only one of the two spaces, to clearly isolate the impact that repulsion in f-space and λ-space has, respectively. The fully-factorized variant has the advantage that only one-dimensional distributions need to be approximated with KDE. However, this also means that correlations between function values f at different points get lost, which will typically result in too strong repulsion. To counteract this issue, it has proven effective to normalize the repulsive term with the factor $\frac{1}{\sqrt{N_d}}$ for this model in our experiments. This factor is heuristically motivated in analogy with the factor that appears in the elementary formula for the standard deviation. In DE solutions, uncertainty at a given point is also influenced by all available measurements. Empirically, we have found this scaling factor to produce excellent results.

5 Experiments

In this section, we present the results obtained from different experiments to evaluate the performance of the approach. The models were trained on noisy data,

$$y(x) = f(x) + \epsilon, \tag{11}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_f^2)$. Ensemble predictions are made by taking the ensemble mean, $\hat{f}(x) = \frac{1}{N_e} \sum_{i=1}^{N_e} \hat{f}_i(x)$, and $\hat{\lambda} = \frac{1}{N_e} \sum_{i=1}^{N_e} \hat{\lambda}_i$, where N_e is the number of ensemble members. Details about the network architecture and training procedure are provided in Appendix E.1. The code for the project is available on GitHub [1].

To evaluate the accuracy of the models, both in terms of the accuracy of the ensemble predictions, as well as the uncertainties, we employ a variety of evaluation metrics. The root mean square error between the

[1] (Link removed for anonymization)

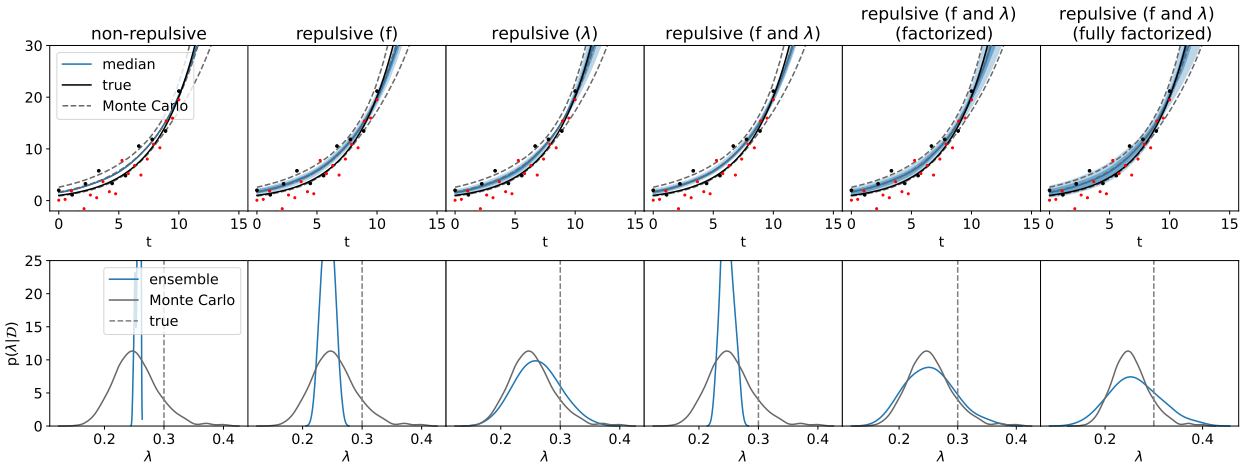

Figure 2: Solving the exponential differential equation (12) with repulsion in (f, λ)-space. **Top row:** The predictions in f-space are depicted. The light- and dark-shaded areas give the regions between the [0.1, 0.9]- and the [0.25, 0.75]-quantiles, respectively. The dashed lines depict the 0.1- and 0.9- quantiles as obtained via MCMC, together with the median. Black dots correspond to train data and red dots to test data. **Bottom row:** The predictions in λ-space are depicted. The ensemble distributions are compared to the distribution obtained via MCMC. The ensembles have 50 members.

Repulsion	RMSE (true)	logL/N (test)	$\lvert\lambda - \hat{\lambda}\rvert$	logL λ	W (f)	W (λ)
non-repulsive	0.62±0.27	-60564.5±93742.0	0.029±0.011	-541.58±969.23	2.5±0.2	0.030±0.001
repulsive (f)	0.69±0.27	-26.0±8.2	0.034±0.021	-10.82±11.51	2.1±0.4	0.028±0.004
repulsive (λ)	**0.59**±0.26	-64.7±21.9	**0.024**±0.010	**1.98**±0.13	0.9±0.1	**0.007**±0.002
repulsive (f, λ)	0.66±0.28	-48.0±18.7	0.032±0.018	-3.69±7.69	2.1±0.3	0.026±0.003
repulsive (f, λ) (factorized)	0.65±0.26	-11.7±3.9	0.027±0.017	1.78±0.17	**0.8**±0.4	0.009±0.004
repulsive (f, λ) (fully factorized)	0.61±0.26	**-4.1**±0.8	0.025±0.011	1.79±0.14	0.8±0.1	0.009±0.003
variational inference (B-PINN)	1.06±0.30	-64.7±36.5	0.066±0.013	-7443.58±2528.78	3.9±1.1	0.055±0.014

Table 3: Evaluation metrics when solving the exponential equation (12) with repulsion in (f, λ)-space, averaged over 5 runs with 50 ensemble members. The mean values plus-or-minus one standard deviation are given. The results for one of these runs are depicted in Figure 2.

ensemble prediction and the true curve (RMSE (true)) quantifies the accuracy of the model. To assess the quality of the ensemble uncertainties, we calculate the log-likelihood per data point on the test data (logL/N (test)). The likelihood is calculated from a KDE approximation of the ensemble distributions via the ensemble members. For the DE parameters λ, we calculate the absolute error $|\lambda - \hat{\lambda}|$ with respect to the true value, and logL λ, the log-likelihood of the true value given the ensemble distribution. Furthermore, we compare how the B-PINN approach using variational inference (VI) from Yang et al. (2021) performs on the same evaluation metrics. An in-depth discussion of VI is given in Appendix D.

For the ODE examples, the fact that the solutions are uniquely determined by the initial conditions and ODE parameters provides the opportunity to perform Markov chain Monte Carlo (MCMC) in the corresponding, low-dimensional space. In this way, a highly accurate proxy for the PINN posterior can be obtained. The reason why it can only be considered a proxy lies in the fact that the MC algorithm acts on exact ODE solutions, whereas the PINN includes the ODE as a soft constraint. We then calculated the Wasserstein distance between the MCMC and the ensemble distributions, both, $W(f)$ in function space (pointwise), and $W(\lambda)$ in λ-space. Details on the MC algorithm are given in Appendix C.

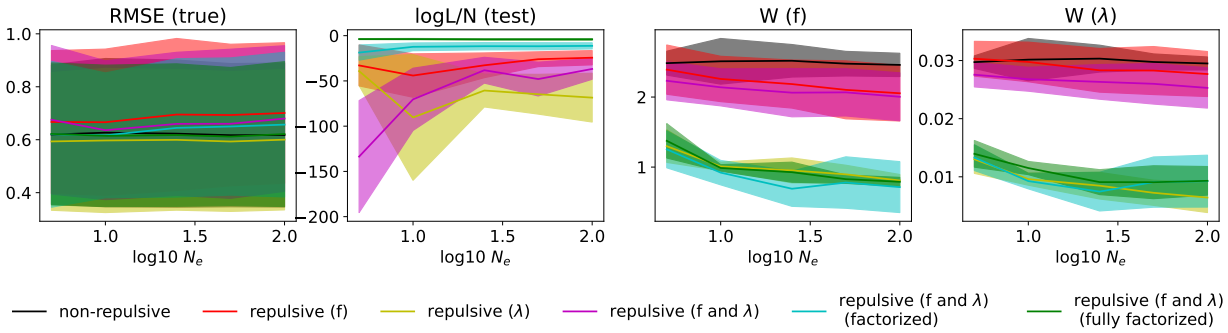

Figure 3: The evaluation metrics when solving the exponential equation with repulsion in (f, λ)-space with different numbers of ensemble members. For better visibility, the non-repulsive ensemble has been omitted for logL/N (test), since the values are far below those of the repulsive ensembles.

5.1 Exponential equation

To illustrate the approach, we consider the example of the exponential differential equation,

$$\frac{\partial f(t)}{\partial t} = \lambda f(t), \tag{12}$$

which has the solution $f(t) = f_0 e^{\lambda t}$. We solve the inverse problem, where λ is determined jointly with the solution $f(t)$. The factor f_0 is obtained automatically when evaluating $f(0)$.

In Figure 2, the results for the different models are depicted, both in terms of the ensemble distribution in f-space, and in λ-space. For the function space, the median of the ensemble is shown together with the 0.1, 0.25, 0.75, and 0.9 quantiles. The corresponding 0.1, 0.9 quantiles and the median as obtained via MCMC are also depicted. The corresponding evaluation metrics are given in Table 3. Plots of the individual ensemble member predictions can be found in Appendix E.3.

It is apparent that the standard, non-repulsive ensemble vastly underestimates the uncertainty in the prediction. The ensemble members almost collapse to a point estimate. In the plot showing the λ distribution, we observe that the predictions are very close to the maximum of the Monte Carlo posterior. This is expected, since standard neural network (and PINN) training converges to MAP solutions. KDE in the high-dimensional space of function values does not seem to be very precise, as indicated by the results for repulsive (f) and repulsive (f and λ), which still underestimate the uncertainties significantly. The models with separate repulsion in λ-space perform notably better. The factorized model in particular adheres well to the MCMC posterior, both in f- and λ-space. The fully-factorized model still agrees well in f-space, but tends to overestimate the uncertainty in λ-space.

In terms of the evaluation metrics, the non-repulsive ensemble performs worst on all metrics apart from RMSE (true) and $|\lambda - \hat{\lambda}|$, which do not measure the uncertainty in the predictions but the accuracy of the mean. The repulsive ensembles are on par with the non-repulsive ensemble when it comes to accuracy, and improve in terms of the uncertainty-quantifying metrics. Whereas the repulsive (λ) model performs best on the metrics involving λ, the factorized and fully-factorized models perform better in function space. The fully-factorized model achieves the highest likelihood on the test data. VI performs clearly worse than the repulsive ensembles.

In Figure 3, evaluation metrics are given for the different models as a function of the number of ensemble members. For metrics that quantify uncertainty, there is a clear tendency for the different variants of repulsive ensembles to improve with the number of ensemble members. This is consistent with the fact that the approximation of the density ρ via KDE becomes more accurate as the number of samples increases. In the limit of infinite ensemble members, the repulsive (f and λ) model is expected to give the true posterior, i.e., $W(f)$ and $W(\lambda)$ should go to values close to zero. For the RMSE, on the other hand, the curves remain

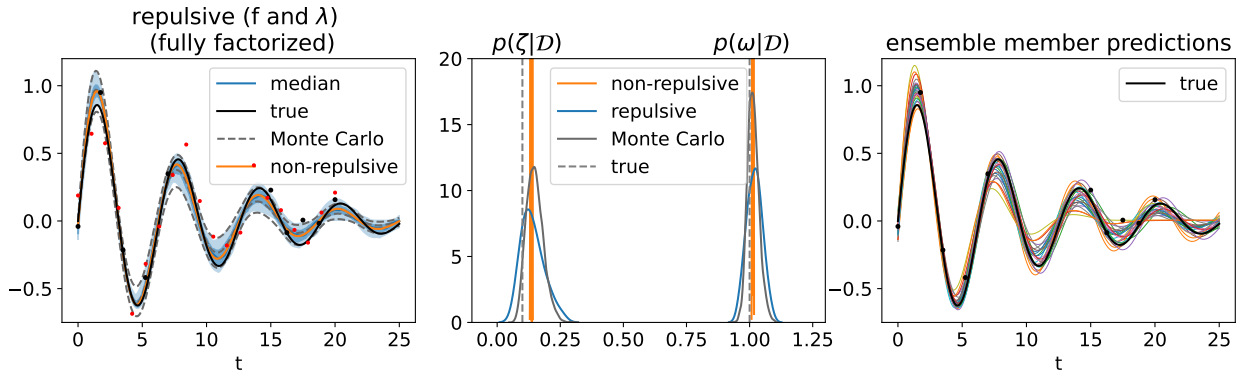

Figure 4: Solving the damped harmonic oscillator (13) with repulsion in $(f,\ \lambda)$-space, using the repulsive ensemble with fully-factorized density. **Left:** The predictions in f-space are depicted. The light- and dark-shaded areas give the regions between the [0.1, 0.9]- and the [0.25, 0.75]-quantiles, respectively. The dashed lines depict the 0.1- and 0.9- quantiles as obtained via MCMC, together with the median. The non-repulsive ensemble effectively collapses to the MAP estimate. Black dots correspond to train data and red dots to test data. **Middle:** The predictions in λ-space are depicted. The ensemble distribution is compared to the distribution obtained via MCMC. **Right:** The individual ensemble members are depicted. The ensembles have 25 members.

Repulsion	RMSE (true)	logL/N (test)	$\lvert\lambda - \hat{\lambda}\rvert$	logL λ	W (f)	W (λ)
non-repulsive	**0.059**±0.013	-1498.9±822.7	**0.016**±0.004	-2903.7±2016.4	0.045±0.004	0.024±0.006
repulsive (f)	0.059±0.014	-287.8±234.2	0.016±0.004	-65.5±62.8	0.041±0.005	0.022±0.006
repulsive (λ)	0.062±0.017	-17.3±3.3	0.021±0.007	**4.7**±0.4	0.027±0.004	**0.010**±0.004
repulsive (f, λ)	0.060±0.014	-860.5±841.0	0.017±0.004	-263.8±247.6	0.043±0.004	0.023±0.006
repulsive (f, λ) (factorized)	0.063±0.017	-13.4±1.1	0.021±0.007	4.7±0.4	0.026±0.004	0.010±0.004
repulsive (f, λ) (fully factorized)	0.061±0.016	**-2.0**±1.5	0.020±0.006	4.6±0.3	**0.025**±0.005	0.013±0.003
variational inference (B-PINN)	0.131±0.035	-116.2±82.3	0.075±0.024	-239285.0±158340.7	0.083±0.022	0.087±0.034

Table 4: Evaluation metrics when solving the damped harmonic oscillator (13) with repulsion in (f, λ)-space, averaged over 5 runs with 25 ensemble members. The mean values plus-or-minus one standard deviation are given. The results for one of these runs are depicted in Figure 4.

essentially flat. This also makes sense, since the RMSE quantifies the accuracy of the mean prediction instead of the uncertainty.

5.2 The damped harmonic oscillator

The DE for the damped harmonic oscillator is given by

$$\frac{\partial f(t)^2}{\partial t^2} + 2\zeta\omega\frac{\partial f(t)}{\partial t} + \omega^2 f(t) = 0, \tag{13}$$

where ω denotes the undamped angular frequency and ζ the damping ratio. In the underdamped regime, where $\zeta < 1$, the solution is $f(t) = f_0 e^{-\zeta\omega t}\sin(\omega\sqrt{1-\zeta^2}t)$.

The results are given in Figure 4 and Table 4. A Figure depicting the ensemble distributions for all the different models is given in Appendix E.4. Again, the non-repulsive ensemble basically collapses to a single solution. The repulsive model, on the other hand, approximates the MC baseline well overall. Note that there is a gap in the data in the area $t \in [7, 15]$; instead of a big blob that unconstrained methods might yield as the uncertainty estimate in this area, the uncertainty intervals, as obtained via the ensemble, retain information about the solution space of the DE, since it is inherent in the ensemble members.

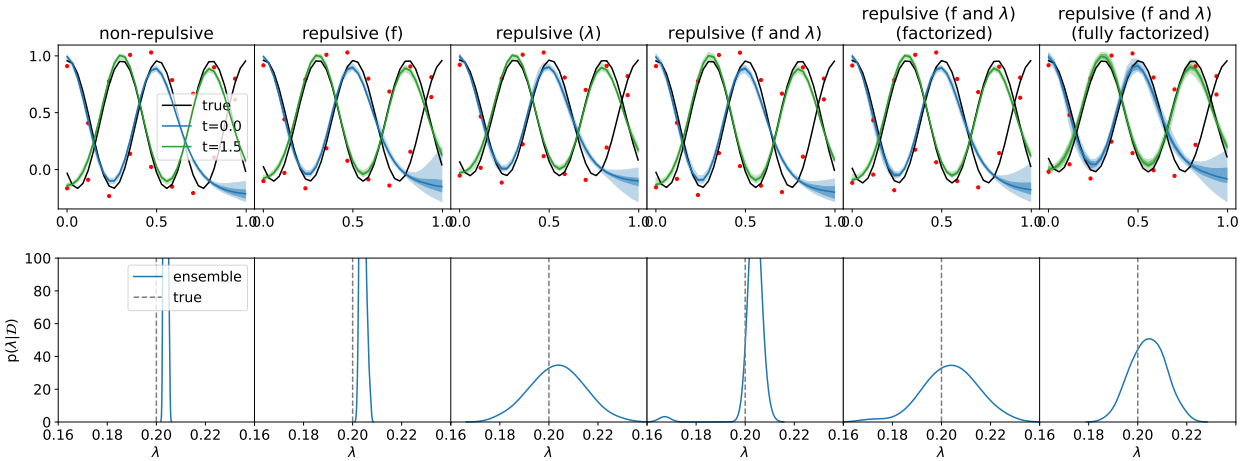

Figure 5: Solving the advection equation from (14) with repulsion in (f, λ)-space. The ensembles have 25 members. Red dots correspond to test data.

Repulsion	RMSE (true)	logL/N (test)	$\lvert\lambda - \hat{\lambda}\rvert$	logL λ
non-repulsive	0.125±0.062	-85.8±79.7	0.020±0.030	-47.4±84.6
repulsive (f)	0.122±0.058	-36.3±31.1	0.020±0.030	-9.9±22.4
repulsive (λ)	0.126±0.064	-44.0±33.1	0.020±0.031	3.5±0.2
repulsive (f and λ)	0.123±0.061	-33.6±15.7	0.018±0.029	-22.3±45.1
repulsive (f and λ) (factorized)	0.121±0.058	-40.3±32.1	0.020±0.031	3.5±0.2
repulsive (f and λ) (fully factorized)	0.121±0.057	**-3.3**±3.1	0.020±0.031	**3.8**±0.6
variational inference (B-PINN)	**0.111**±0.017	-31.4±11.5	**0.005**±0.003	-392.6±409.4

Table 5: Evaluation metrics when solving the advection equation (14) with repulsion in (f, λ)-space, averaged over 5 runs with 25 ensemble members. The mean values plus-or-minus one standard deviation are given. The results for one of these runs are depicted in Figure 5.

When considering the posteriors for the DE parameters, we again find that the non-repulsive ensemble assigns exceedingly high certainty to values that are close to, but not exactly at, the true value. The repulsive ensemble posterior again agrees reasonably well with the MC posterior. In the rightmost plot, the individual ensemble members are shown. Each of them constitutes a PINN solution in its own right. While close to real solutions of the ODE, there can be small deviations due to the soft-constraint nature of the PINN.

The evaluation metrics in Table 4 show a very similar pattern to those in the previous example. On the whole, the fully-factorized model performs best.

5.3 The advection equation

The advection equation is given by

$$\frac{\partial u(t,x)}{\partial t} + \lambda \frac{\partial u(t,x)}{\partial x} = 0. \tag{14}$$

The solution for this PDE is given by $u(t,x) = u_0(x - \lambda t)$, where u_0 denotes the initial conditions.

An illustration of the datasets under consideration is given in Figure 6. The data was obtained from Takamoto et al. (2022). The results are given in Figure 5 and Table 5. The solutions in function space are illustrated along two cuts of the 2D domain. In function space, the difference between non-repulsive and repulsive ensembles is smaller than in the previous examples, but the non-

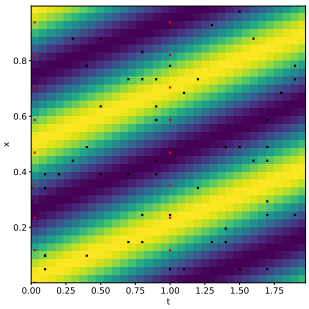

Figure 6: Example solution of the advection equation. Black dots correspond to training data and red dots to test data.

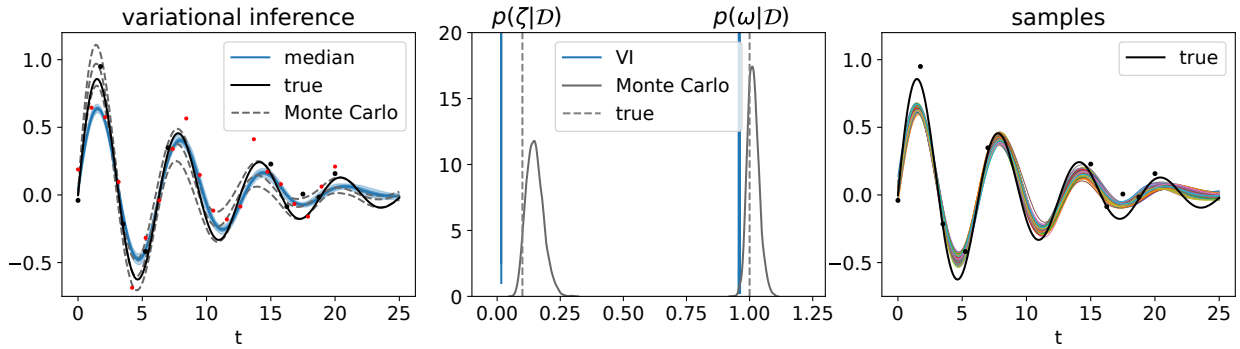

Figure 7: Solving the damped harmonic oscillator (13) with variational inference. **Left:** The predictions in f-space are depicted. The light- and dark-shaded areas give the regions between the [0.1, 0.9]- and the [0.25, 0.75]-quantiles, respectively. The dashed lines depict the 0.1- and 0.9-quantiles as obtained via MCMC, together with the median. Black dots correspond to train data and red dots to test data. **Middle:** The predictions in λ-space are depicted. The VI distribution is compared to the distribution obtained via MCMC. **Right:** Samples from the VI posterior are depicted.

repulsive ensemble still tends to underestimate uncertainty, especially in regions without data (see the right part of the $t = 0.0$ curve).

For this example, no MC solution is available, due to the continuous boundary condition. Nevertheless, from Table 5, we see that the repulsive ensembles perform significantly better on the log-likelihoods than the non-repulsive ensemble. Altogether, the fully-factorized model again performs best.

5.4 Comparison with variational inference

In this section, we evaluate how VI performs as a method for Bayesian inference in PINNs, using the B-PINN from Yang et al. (2021). An in-depth discussion of the method is given in Appendix D.

In Figure 7, the performance of VI applied to the damped harmonic oscillator is illustrated. The mean of the distribution follows the MC median reasonably well. When it comes to sample diversity, VI performs clearly worse than the RE-PINN (compare Figure 4). The posterior distributions for the DE parameters are strongly squeezed to specific values, presumably because the mean-field Gaussian posterior does not allow for correlations between λ and θ, and this is the only way for the model to attain reasonable DE fulfillment. The individual samples do not adhere to the DE consistently either. For example, the maximum amplitude curve should be the same at each extremum, but this is not the case. Instead, they mostly appear to be slightly shifted versions of the mean prediction.

From the results in Tables 3-5, it is apparent that VI performs worse in terms of uncertainty quantification than the RE-PINN. While it does improve over the non-repulsive ensemble in terms of the log-likelihood on the test data, the log-likelihood on the DE parameters is significantly worse, indicating strongly squeezed posteriors. Furthermore, the improvement in uncertainty quantification over the non-repulsive ensemble often comes at the cost of a degradation in the mean predictions as measured by RMSE (true) and $|\lambda - \hat{\lambda}|$. A degradation in the sample quality is also reflected in higher mean values of the DE residuals, which are given in Table 6 in the appendix. The likely reason for this is that the variational posterior does not include correlations between model parameters. While better-suited posteriors could likely be designed, this would come at the cost of significant amounts of additional model tuning.

Another drawback of VI is that the width of the uncertainty intervals remains arbitrary, since the weighting factor of the KL term between prior and posterior in the VI loss function (25) needs to be hand-tuned. The reason for this is that the prior put on the network parameters has no obvious relation to function space, and should therefore not be given too much weight. In total, we conclude that VI does not yield truly Bayesian posteriors.

6 Conclusions

In this work, we introduced the RE-PINN, and investigated the application of repulsive ensembles for uncertainty estimation to PINNs. Building on the work of D'Angelo and Fortuin (2021), we derived an expression for the repulsive term in PINNs and showed that DE parameters need to be included in the repulsion. The repulsion results in an ensemble distribution that accurately represents the Bayesian posterior in the limit of many ensemble members. We investigated different approximations of this repulsive term and found that factorizing the density into lower-dimensional components improves the performance of the repulsive ensemble.

Our experiments demonstrate that the standard ensemble vastly underestimates the uncertainty in PINN predictions. The reason for this presumably lies in the fact that the PINN loss acts as an additional regularizer that further discourages diversity in the ensemble members, beyond the fact that the ensemble members converge to MAP estimates. Including repulsion is therefore essential in the PINN context and significantly improves the results. In accordance with the theoretical derivation from Section 4, applying repulsion in the joint space of function values f and DE parameters λ works best, with the fully-factorized model giving the most consistently good results. The model predictions agree well with the Monte Carlo baselines, where available, confirming that repulsive ensembles can be considered Bayesian.

Our approach also outperformed B-PINN, the VI approach introduced by Yang et al. (2021), by a significant margin. The latter is lacking in sample diversity, yields arbitrary widths of the uncertainty intervals, and often exhibits a degradation in the sample quality as compared to the MAP estimate. The RE-PINN, on the other hand, is well equipped to combine both sample diversity and sample accuracy. Since each of the ensemble members has its own set of parameters, distinct solutions can be learned accurately. In the RE-PINN, a principled way for weighting the repulsive term is also available, leading to uncertainty estimates in agreement with the Bayesian posterior. These properties make repulsive ensembles a natural choice for uncertainty estimation in PINNs.

In the future, it would be interesting to further explore possible ways of approximating the ensemble density. Using Gaussian processes to approximate the ensemble density $\rho(f)$ could take correlations between values in f-space into account effectively and may reduce the number of ensemble members necessary for obtaining good results. The approach of Zou et al. (2025) could presumably also be improved with repulsion, as it would encourage the differently initialized PINNs to converge to different DE solutions. It would also be interesting to experiment with different methods of repulsion, such as repulsion in the space of input-gradients (Trinh et al., 2024), instead of the function values. Investigating ways to enable sequential training of the ensemble members constitutes another promising direction of research. It might be possible to store particle trajectories of preceding runs in a low-dimensional space, e.g., in (f, λ)-space, and use these for repulsion when training the remaining ensemble members.

References

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and whatâĂŹs next. *Journal of Scientific Computing*, 92(3):88, 2022.

Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021a.

Zaharaddeen Karami Lawal, Hayati Yassin, Daphne Teck Ching Lai, and Azam Che Idris. Physics-informed neural network (PINN) evolution and beyond: A systematic literature review and bibliometric analysis. *Big Data and Cognitive Computing*, 6(4):140, 2022.

Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021b.

George EP Box and George C Tiao. *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on Bayesian neural networks–A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11): 2, 2011.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, pages 9030–9039. PMLR, 2021.

Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function Space Particle Optimization for Bayesian Neural Networks. In *International Conference on Learning Representations*, 2019.

Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are Bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Peng Li, Dario Grana, and Mingliang Liu. Bayesian neural network and Bayesian physics-informed neural network via variational inference for seismic petrophysical inversion. *Geophysics*, 89(6):1–46, 2024.

Xinchao Jiang, Xin Wang, Ziming Wen, Enying Li, and Hu Wang. Practical uncertainty quantification for space-dependent inverse heat conduction problem via ensemble physics-informed neural networks. *International Communications in Heat and Mass Transfer*, 147:106940, 2023.

Francisco Sahli Costabal, Yibo Yang, Paris Perdikaris, Daniel E Hurtado, and Ellen Kuhl. Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, 8:42, 2020.

Katsiaryna Haitsiukevich and Alexander Ilin. Improved training of physics-informed neural networks with model ensembles. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

Zongren Zou, Zhicheng Wang, and George Em Karniadakis. Learning and discovering multiple solutions using physics-informed neural networks with random initialization and deep ensemble. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2325), 2025.

Zhiwei Fang, Sifan Wang, and Paris Perdikaris. Ensemble learning for physics informed neural networks: A gradient boosting approach. *arXiv preprint arXiv:2302.13143*, 2023.

Mingyuan Yang and John T Foster. Multi-output physics-informed neural networks for forward and inverse PDE problems with uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 402:115041, 2022.

Hai Siong Tan, Kuancheng Wang, and Rafe McBeth. Evidential physics-informed neural networks. In *Proceedings of the International Conference on Scientific Computing and Machine Learning (SCML 2025)*, pages 22–31, 2025.

Lennart Röver, Björn Malte Schäfer, and Tilman Plehn. PINNferring the Hubble function with uncertainties. *energy*, 5:9, 2024.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

Trung Trinh, Markus Heinonen, Luigi Acerbi, and Samuel Kaski. Input-gradient space particle inference for neural network ensembles. In *The Twelfth International Conference on Learning Representations*, 2024.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

Matthew D Hoffman, Andrew Gelman, et al. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Background

Additional details on methods and results utilized in the main paper are given in this appendix.

A.1 Derivation of the repulsive term

In this section, we outline the arguments that were employed in Section 3.2 of D'Angelo and Fortuin (2021) to derive the repulsive term in (5).

They consider the space of probability measures $\mathcal{P}_2(\mathcal{M})$ on the manifold $\mathcal{M}$,

$$\mathcal{P}_2\mathcal{M} = \left\{ \varphi : \mathcal{M} \to [0, \infty) \middle| \int_{\mathcal{M}} d\varphi = 1, \int_{\mathcal{M}} |x|^2 \varphi(x) dx < +\infty \right\}, \tag{15}$$

together with the Wasserstein metric

$$W_2^2(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int |x - y|^2 d\pi(x, y). \tag{16}$$

They then proceed to employ the KL divergence to match the ensemble density $\rho(x)$ to the target distribution $\pi(x)$:

$$\inf_{\pi \in \mathcal{P}_2\mathcal{M}} D_{KL}(\rho, \pi) = \int_{\mathcal{M}} \big(\log \rho(x) - \log \pi(x) \big) \rho(x) dx. \tag{17}$$

When minimizing the KL divergence, the evolution of the measure ρ (also referred to as the Wasserstein gradient flow), is given by the Liouville equation, which in turn simplifies to the Fokker-Planck equation:

$$\frac{\partial \rho(x)}{\partial t} = \nabla \cdot \Big(\rho(x) \nabla \big(\log \rho(x) - \log \pi(x) \big) \Big) \tag{18}$$

$$= -\nabla \cdot (\rho(x) \nabla \log \pi(x)) + \Delta \rho(x). \tag{19}$$

This equation admits $\pi(x)$ as the unique stationary solution. The corresponding deterministic particle dynamics ODE, for samples $x_i \sim \rho$, is then given by

$$\frac{dx_i}{dt} = -\nabla \big(\log \rho(x_i) - \log \pi(x_i) \big), \tag{20}$$

or, after discretizing it,

$$x_i^{t+1} = x_i^t + \epsilon_t \left(\nabla \log \pi(x_i^t) - \nabla \log \rho(x_i^t) \right). \tag{21}$$

In the context of repulsive ensembles, ρ denotes the ensemble distribution and $\pi(x)$ is chosen as the posterior $p(x|\mathcal{D})$. Note that, during the derivation, no restrictions have been placed on the domain of $\rho(x)$. Hence, x can be chosen as the network parameters θ, or the function values $\boldsymbol{f}$, depending on which quantity is of interest. This also vindicates us in writing down (6) when considering $\rho(\boldsymbol{f}, \lambda)$ instead of $\rho(\boldsymbol{f})$.

B Kernel density estimation

Given samples x_i, the density $\rho(x)$ can be approximated using kernel density estimation (KDE). We then obtain the following expression:

$$\rho(x) \approx \frac{1}{\sqrt{|h|} N_e} \sum_{i=1}^{N_e} k \left(\frac{x - x_i}{h} \right), \tag{22}$$

where h is a vector containing the bandwidth for each dimension of x; the division in the argument of the kernel is elementwise. The bandwidths are estimated with the median heuristic (Liu and Wang, 2016), $h_d = \frac{\text{med } r_{ij}^d}{\sqrt{\log N_e}}$, where r_{ij}^d denotes the pairwise distances between samples along the given dimension d. We use the RBF kernel, $k(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\|z\|^2}{2}}$. The case of joint densities $\rho(y, z)$ can be reduced to the given case via $x = \begin{bmatrix} y \\ z \end{bmatrix}$.

C Monte Carlo estimates of the posterior

The true posterior, $p(f, \lambda|\mathcal{D}) = \frac{p(\mathcal{D}|f,\lambda)p(f,\lambda)}{p(\mathcal{D})}$, is in general intractable. In the case of ODEs, however, the solution is uniquely determined by the initial conditions $\{f_{0j}\}_{j=0}^{d}$ and the ODE parameters $\{\lambda_j\}_{j=0}^{N_\lambda}$. Hence, it is possible to conduct Markov Chain Monte Carlo in the resulting $(d + N_\lambda)$-dimensional space to obtain samples from $p(f_0, \lambda|\mathcal{D})$, which can serve as a proxy for the PINN posterior.

For example, in the case of the exponential equation (12), the current state of the chain would be defined by (f_0, λ). The corresponding solution $f(t) = f_0 e^{\lambda t}$ can be employed to calculate the likelihood $p(\mathcal{D}|f_0, \lambda)$. Together with the prior $p(f_0, \lambda) = p(f_0)p(\lambda)$, all of the quantities required for evaluating a proposal are available.

For the MCMC algorithm, we made the following choices of priors: for the exponential equation from Section 5.1, we chose $p(f_0) \sim \mathcal{U}(-10, 10)$ and $p(\lambda) \sim \mathcal{U}(-10, 10)$. For the damped harmonic oscillator from Section 5.2, we chose $p(f_0) \sim \mathcal{U}(0, 1.5)$, $p(\omega) \sim \mathcal{U}(0, 3)$, and $p(\zeta) \sim \mathcal{U}(0, 0.9)$. When training the PINNs, the same priors can be used by plugging them into (10). However, to ensure stable training with useful gradients, it is necessary to use smoothened versions of the priors, i.e., to use Sigmoids instead of step functions.

We used the pyro package (Bingham et al., 2018) for the implementation of the Monte Carlo algorithm, and chose Hamiltonian Monte Carlo (HMC) (Neal et al., 2011) with the No-U-Turn Sampler (NUTS) (Hoffman et al., 2014).

D Variational inference

We conduct VI as described in Yang et al. (2021), as a variant of the B-PINN. Here, the network parameters are not fixed weights, but instead distributions to be learned. We employ the commonly used mean-field Gaussian approximation, where the variational posterior density is factorized into independent Gaussian distributions: $q(\bar{\theta}; \zeta) = \prod_j q_j(\bar{\theta}_j; \zeta_{\mu,j}, \zeta_{\sigma,j})$, where $\bar{\theta} = \{\theta, \lambda\}$ contains both, the network parameters θ and the DE parameters λ. The means ζ_μ and standard deviations $\ln(1 + \exp(\zeta_\sigma))$ of these Gaussians are the parameters to be optimized during training. The loss function for BNN-VI is given by

$$\mathcal{L}(\zeta) = \frac{1}{N_z} \sum_{i=1}^{N_z} \ln q(\bar{\theta}^{(i)}) - \ln p(\bar{\theta}^{(i)}) - \ln p(\mathcal{D}|\bar{\theta}^{(i)}), \tag{23}$$

$$\approx D_{\mathrm{KL}}(q(\bar{\theta}), p(\bar{\theta})) + \frac{1}{N_z} \sum_{i=1}^{N_z} \mathcal{L}_f(\bar{\theta}^{(i)}) + \mathcal{L}_{\mathcal{F}}(\bar{\theta}^{(i)}), \tag{24}$$

where the $\bar{\theta}^{(i)}$ denote N_z sets of parameters sampled from the variational posterior q. Hence, the term $\frac{1}{N_z}\sum_i \ln q(\bar{\theta}^{(i)}) - \ln p(\bar{\theta}^{(i)}) \approx D_{\mathrm{KL}}(q(\bar{\theta}), p(\bar{\theta}))$ gives an estimate of the KL divergence between the variational posterior and the prior. When assuming Gaussian priors, it can be evaluated analytically. The likelihoods $\mathcal{L}_f$ and $\mathcal{L}_{\mathcal{F}}$ are given in (4). After rescaling, the loss can be written in terms of the same weighting factors as in (10):

$$\mathcal{L}(\zeta) = \frac{1}{N_d N_z} \sum_{i=1}^{N_z} \sum_{j=1}^{N_d} \left(\hat{f}_i(x_j) - y_j\right)^2 + \frac{f_\lambda}{N_c N_z} \sum_{i=1}^{N_z} \sum_{j=1}^{N_c} \left(\mathcal{F}(\hat{f}_i(x_j), \lambda_i)\right)^2 + f_{\mathrm{KL}} D_{\mathrm{KL}}(q(\bar{\theta}), p(\bar{\theta})). \tag{25}$$

The factor f_{KL} constitutes an ad hoc weighting factor for the KL term, which was required for obtaining reasonable results; omitting it led to standard deviations in the NN parameters that were too large and which led to nonsensical predictions that did not fit the data nor the DE well.

The performance of VI could likely be improved by choosing a more sophisticated posterior distribution. However, determining a suitable choice would introduce significantly more tuning work than in the RE-PINN. Another advantage of the RE-PINN over VI is that inference can be conducted directly in function space. For VI, priors need to be defined in weight space, and it is not obvious what a good choice would be. In our experiments, we chose zero-mean Gaussian priors with standard deviation 0.1 for all of the network parameters.

Model	exponential equation	damped HO	advection	Lotka-Volterra
non-repulsive	**0.073**±0.023	**0.011**±0.002	0.055±0.007	0.330±0.021
repulsive (f)	0.135±0.017	0.013±0.002	0.055±0.007	0.359±0.061
repulsive (λ)	0.089±0.019	0.014±0.002	0.062±0.006	**0.313**±0.035
repulsive (f and λ)	0.116±0.016	0.111±0.002	0.055±0.008	0.335±0.067
repulsive (f and λ) (factorized)	0.123±0.017	0.013±0.002	0.062±0.006	0.359±0.054
repulsive (f and λ) (fully factorized)	0.080±0.021	0.012±0.002	**0.054**±0.007	0.399±0.076
B-PINN/BNN-VI	0.220±0.026	0.037±0.004	0.077±0.016	0.816±0.081

Table 6: Adherence of individual posterior samples to the DE, as measured by the mean DE residuals $\frac{1}{N_z}\sum_{i=1}^{N_z} 2\sigma_{\mathcal{F}}^2 \mathcal{L}_{\mathcal{F}}^i$. Repulsion does not significantly impact the quality of individual samples. For samples from the VI posterior, the DE fulfillment is adversely affected.

E Experiments

In this appendix, we give background information about the experiments conducted in Section 5 and present an additional experiment.

E.1 Training details

Fully-connected neural networks with tanh activation function have been employed for the individual ensemble members. To better isolate the impact of repulsion on the training outcome, the ensemble members were initialized with the same values for the different models given in Table 2. The Adam optimizer (Kingma and Ba, 2014) was used for all of the examples. When training PINNs, it is often advantageous to start with a lower weight for the DE constraint and only increase it later in the training. Similarly, only activating the repulsion later in the training can lead to improved KDE estimates and better convergence.

For the exponential equation (5.1), the networks had 2 layers of width 20 and the training lasted 10 000 iterations with learning rate 0.01. The strength of the measurement noise was $\sigma_f = 2$. The weighting factor f_λ in (10) was chosen as 5 and then increased to 10 and 25 at iterations 7 500 and 9 000, respectively. For the damped harmonic oscillator (5.2), 3 layers of width 20 were chosen and the training lasted 15 000 iterations with learning rate 0.01. The strength of the measurement noise was $\sigma_f = 1$. The weighting factor f_λ in (10) was chosen as 1 and then increased to 5 and 10 at iterations 5 000 and 7 500, respectively. For this example, the repulsive term was only activated after 3 000 iterations. For the advection equation (5.3), 4 layers of width 100 were chosen and the training lasted 5 000 iterations with learning rate 0.003, with weight decay set to 5×10^{-4}. The strength of the measurement noise was $\sigma_f = 0.1$. The weighting factor f_λ in (10) was chosen as 0.1. For the Lotka-Volterra equation (E.6), 3 layers of width 20 were chosen and the training lasted 15 000 iterations with learning rate 0.01. The strength of the measurement noise was $\sigma_f = 1$. The weighting factor f_λ in (10) was chosen as 0.5 and then increased to 1 and 2 at iterations 5 000 and 7 500, respectively.

When training BNN-VI (Appendix D), the same network architectures were employed. For the exponential equation (12), the training lasted 10 000 iterations with learning rate 0.01. The weighting factor f_λ was initialized as 1 and then increased to 2 and 10 after 1 000 and 5 000 iterations, respectively. The KL term was activated after 5000 iterations with a weighting factor of $f_{\mathrm{KL}} = 0.01$. For the damped harmonic oscillator (13), the training lasted 10 000 iterations with learning rate 0.01. The weighting factor f_λ was initialized as 1 and then increased to 2 and 10 after 2000 and 3 000 iterations, respectively. The KL term was activated after 3000 iterations with a weighting factor of $f_{\mathrm{KL}} = 0.00002$. For the advection equation (14), the training lasted 15 000 iterations with learning rate 0.003. The weighting factor f_λ was initialized as 0.01 and then increased to 0.1 after 10 000 iterations. The KL term was activated after 10000 iterations with a weighting factor of $f_{\mathrm{KL}} = 10^{-7}$. For the Lotka-Volterra equation (26), the training lasted 10 000 iterations with learning rate 0.01. The weighting factor f_λ was initialized as 0.1 and then increased to 1 and 2 after 2 000 and 5 000 iterations, respectively. The KL term was activated after 5 000 iterations with a weighting factor of $f_{\mathrm{KL}} = 0.00001$.

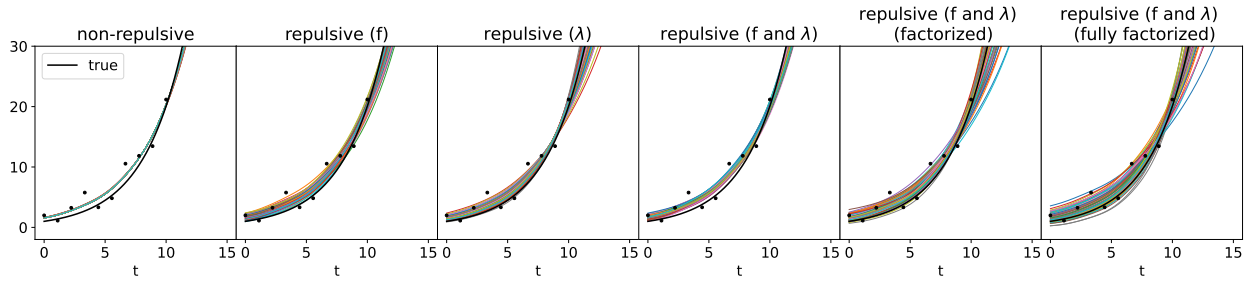

Figure 8: The ensemble members corresponding to the distributions in Figure 2 are depicted.

E.2 Computational resources

The experiments were conducted on a computer with NVIDIA GeForce Titan XP 12GB GPU and Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz CPU. The runtimes for the different experiments are given in Table 7. Adding repulsion does not increase the runtime by a large amount. The different repulsive models have very similar runtimes and are therefore not listed individually.

Table 7: Runtime (in hours) for one run contributing to the entries in Tables 3-8.

	non-repulsive	repulsive
exponential equation (5.1)	0.35	0.43
damped harmonic oscillator (5.2)	0.69	0.78
advection equation (5.3)	0.39	0.46
Lotka-Volterra (E.6)	0.69	0.78

E.3 Exponential equation

In Figure 8, the individual ensemble members corresponding to Figure 2 are depicted. In the plots, it can be observed that these curves adhere to the ODE individually, i.e. solutions that start at a higher initial value learned a lower value for the corresponding exponent λ in the solution of (12) and eventually cross through other samples, which start at lower initial values but with larger λ. This behavior also leads to the narrowing of the uncertainty intervals around $t = 10$, observed in Figure 2.

E.4 Damped harmonic oscillator

Figure 9 shows results for the damped harmonic oscillator corresponding to Figure 4 and Table 4 for all of the models.

E.5 Advection equation

In Figure 10, individual members of the ensembles corresponding to the results in Section 5.3 are depicted. Also here, the different ensemble members constitute valid PDE solutions independently.

E.6 The Lotka-Volterra equations

The Lotka-Volterra equations describe predator-prey dynamics and are given by

$$\frac{\partial x}{\partial t} = \alpha x - \beta xy, \tag{26a}$$

$$\frac{\partial y}{\partial t} = -\gamma y + \delta xy. \tag{26b}$$

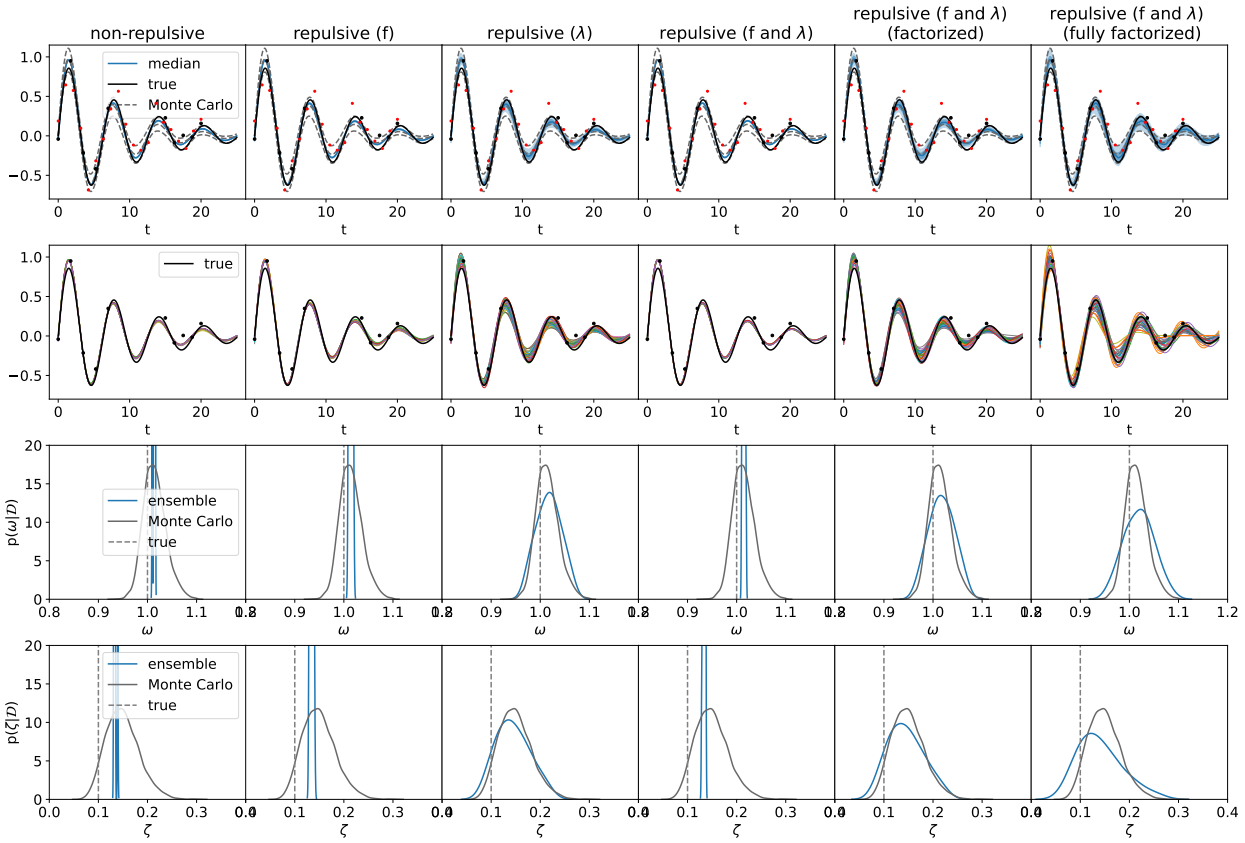

Figure 9: Solving the damped harmonic oscillator (13) with repulsion in (f, λ)-space. **Top row:** The predictions in f-space are depicted. The light- and dark-shaded areas give the regions between the [0.1, 0.9]- and the [0.25, 0.75]-quantiles, respectively. The dashed lines depict the 0.1- and 0.9-quantiles as obtained via MCMC, together with the median. Black dots correspond to train data and red dots to test data. **Bottom row:** The predictions in λ-space are depicted. The ensemble distribution is compared to the distribution obtained via MCMC. The corresponding evaluation metrics are given in Table 4. The ensembles have 25 members.

The results are given in Figures 11 and Table 8. The non-repulsive ensemble members again collapse to very similar solutions. The different repulsive ensembles exhibit higher variation in their predictions and show altogether very similar performance. They improve over the non-repulsive ensemble in terms of logL/N (test), and they yield good values on logL λ more consistently. MCMC did not converge for this example.

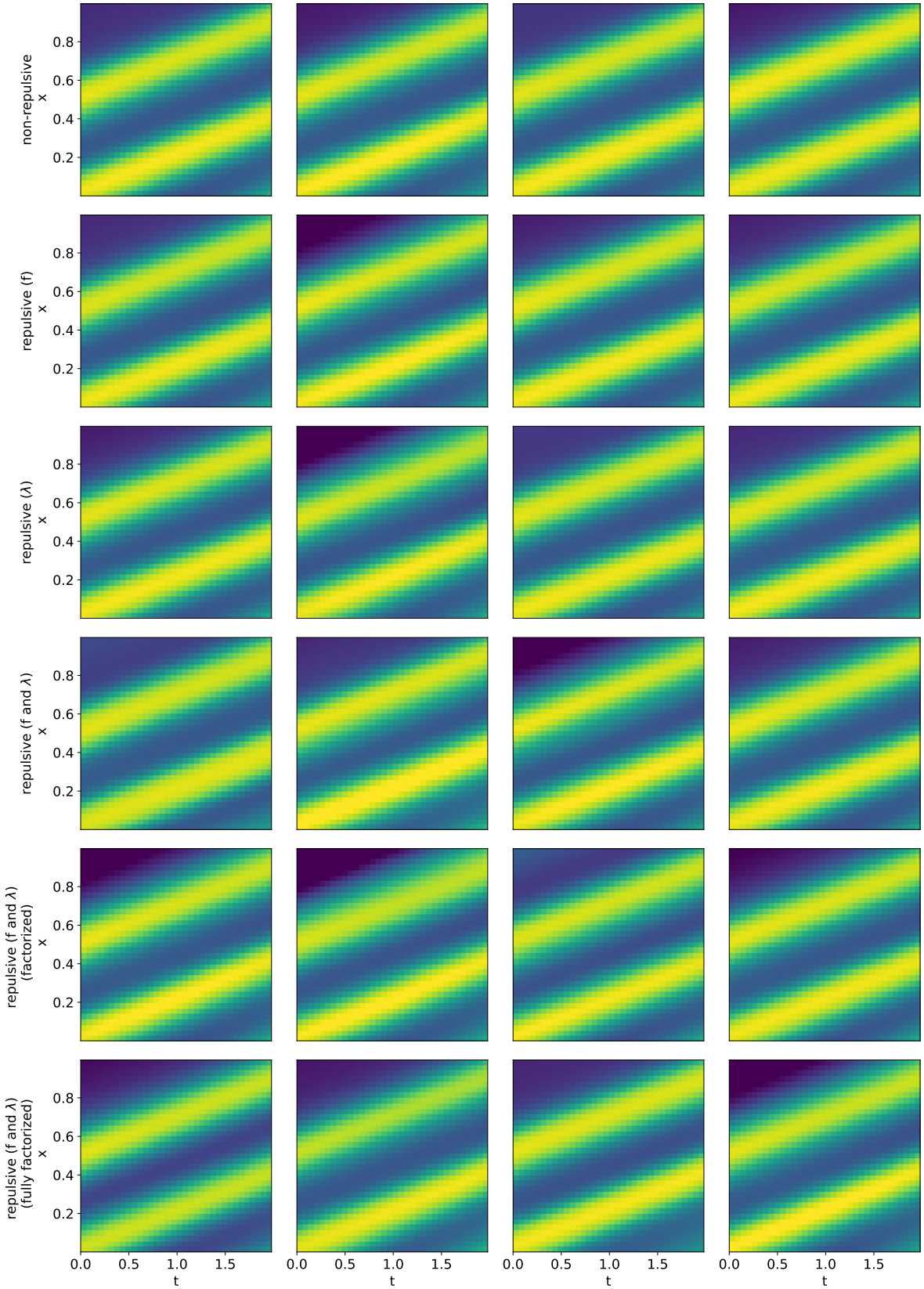

Figure 10: A subset of the ensemble members corresponding to the distributions in Figure 5 is depicted.

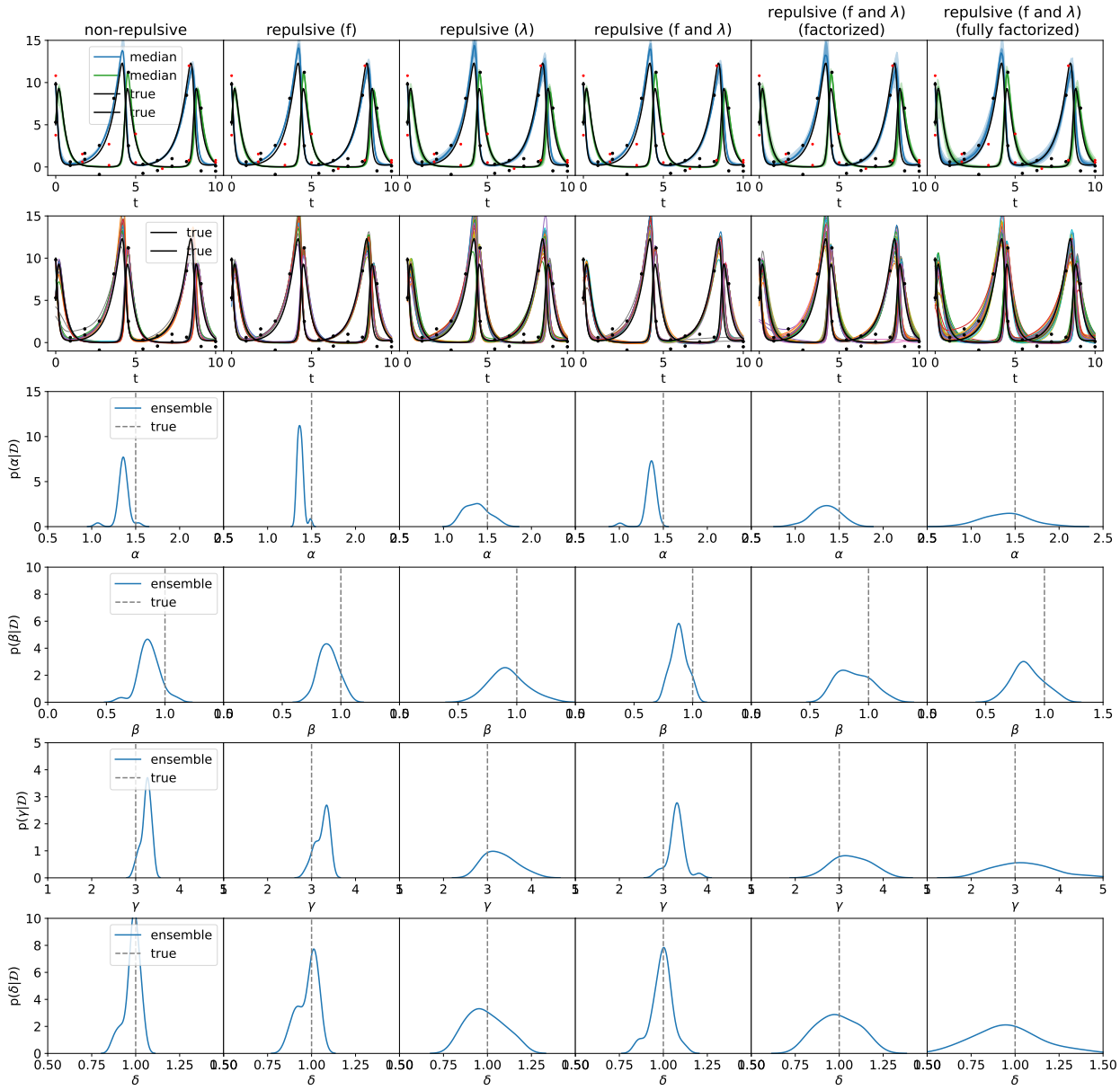

Figure 11: Solving the Lotka-Volterra equations (12) with repulsion in (f, λ)-space. **Top row:** The predictions in f-space are depicted. The light- and dark-shaded areas give the regions between the [0.1, 0.9]- and the [0.25, 0.75]-quantiles, respectively. Black dots correspond to train data and red dots to test data. **Bottom row:** The predictions in λ-space are depicted. The ensembles have 25 members.

	RMSE (true)	logL/N (test)	$\lvert\lambda - \hat{\lambda}\rvert$	logL λ
non-repulsive	0.54±0.10	-720.1±380.5	0.13±0.04	0.98±2.05
repulsive (f)	0.55±0.11	-673.1±575.9	0.13±0.03	-1.88±5.28
repulsive (λ)	**0.52**±0.11	-267.0±151.8	0.11±0.04	**1.92**±0.60
repulsive (f and λ)	0.54±0.10	-825.3±633.6	0.12±0.03	-1.95±4.73
repulsive (f and λ) (factorized)	0.56±0.07	-228.1±133.9	**0.11**±0.04	1.81±0.43
repulsive (f and λ) (fully factorized)	0.54±0.09	**-39.4**±12.2	0.12±0.02	0.15±0.54
B-PINN/BNN-VI	1.96±0.19	-575.4±193.0	0.36±0.07	-7476.72±2000.14

Table 8: Evaluation metrics when solving Lotka-Volterra equation (26) with repulsion in (f, λ)-space, averaged over 5 runs with 25 ensemble members. The mean values plus-or-minus one standard deviation are given. The results for one of these runs are depicted in Figure 11.