6. Results

This chapter is devoted to the presentation of the results obtained in different applications, ranging from the regression task to differential problems such as the Laplace and Damped Harmonic Oscillator models.

On these test cases, we made experiments aimed at showing the main characteristics of the models that our library includes, especially comparing training algorithms among each other (either deterministic and bayesians) and physics-informed versus non-physics-informed learning. We also made trials of solving the same problem with different datasets, in order to estimate possible issues linked to the presence of more complex functions.

The organization of the sections reflects different aspects captured by our implementation: indeed, we focus separately on a comparison among training algorithms on the same task (subsection 6.1), on the implications of physics-informed learning (subsection 6.2) and on code portability to higher dimension (subsection 6.3). The last section, subsection 6.4, presents some results obtained after an intense tuning of the parameters.

In this phase of result production, indeed, the tuning of parameters had a big input: as stated in subsection 1.1, the focus of the project was the library feature expansion, but we showed as well the possibility to proceed with the optimization of the results in one specific application.

An intensive tuning effort was made with the B-PINN trained to solve the Laplace problem with the HMC training (presented in subsection 6.4): this represented a challenging task, because HMC is the method having the most careful parameters' choice as discussed in subsubsection 3.1.3, but on the other hand it enabled to perform tuning with test cases having a feasible computational time, also thanks to the presence of debug utilities such as the continuous monitoring of the acceptance rate during training.

This needs to be taken into account when reading the following showroom of results, in order to avoid to make hasty comparisons among the methods just basing on the quality of the prediction in terms of errors and UQ.

At the beginning of each test case, we reported the options set to obtain the results shown (that could be retrieved even from the logs in the corresponding `outs` subfolder.

6.1. Regression Problem

This section is devoted to the presentation on the results obtained for the function interpolation task. This case of application falls outside the physics-informed framework, because the model does not learn a PDE during training, but it represented a suitable starting point, both because in this way we got results to make comparisons on the contribution of the physics and because, from the tuning point of view, it represented a faster way to identify a good set of algorithm parameters which enabled to obtain good-quality results.

Moreover, as stated in subsubsection 4.4.1, the regression can be seen as preparatory to the physics-informed case also because we do not let the PDE information enter immediately into the loss used for training, but start including it after having reached a suitable approximation of the measurement data with precisely some regression-only epochs.

In this section, we show for all the methods implemented the reconstruction of the function $u(x) = \cos(8x)$ in the domain $\Omega = [0, 1]$, having $N_{fit} = 16$ equidistributed measurements available which are affected by a noise with level $\sigma_u = 0.1$.

6.1.1. Adam

We first present the results obtained with the Adam method; this training choice is deterministic, hence it cannot output any UQ estimate on the prediction, but it enabled accurate results in an extremely short time, as witnessed by Figure 19.

Thanks to its light computational overload, the Adam algorithm turned out to be a handy initializer for letting the bayesians algorithms start from not completely random initial values.

The number of epoch was set to $N = 5000$ and the training lasted ~ 2 minutes; the Adam-specific parameters were fixed (actually, in all the Adam test cases of the project) to the good default values suggested in [5], that are:

Parameter	Value
β_1	0.9
β_2	0.999
ϵ	10^{-8}
η	10^{-3}

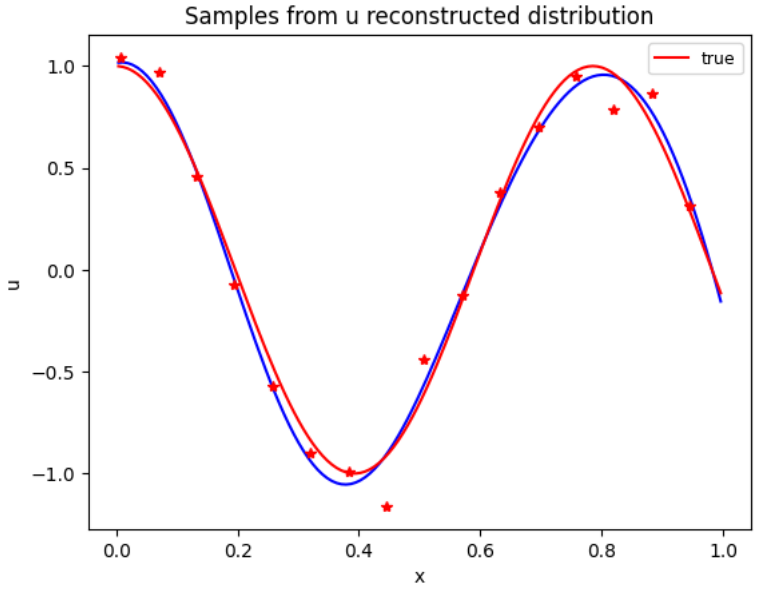


Figure 19: Function regression with Adam on $u(x) = \cos(8x)$

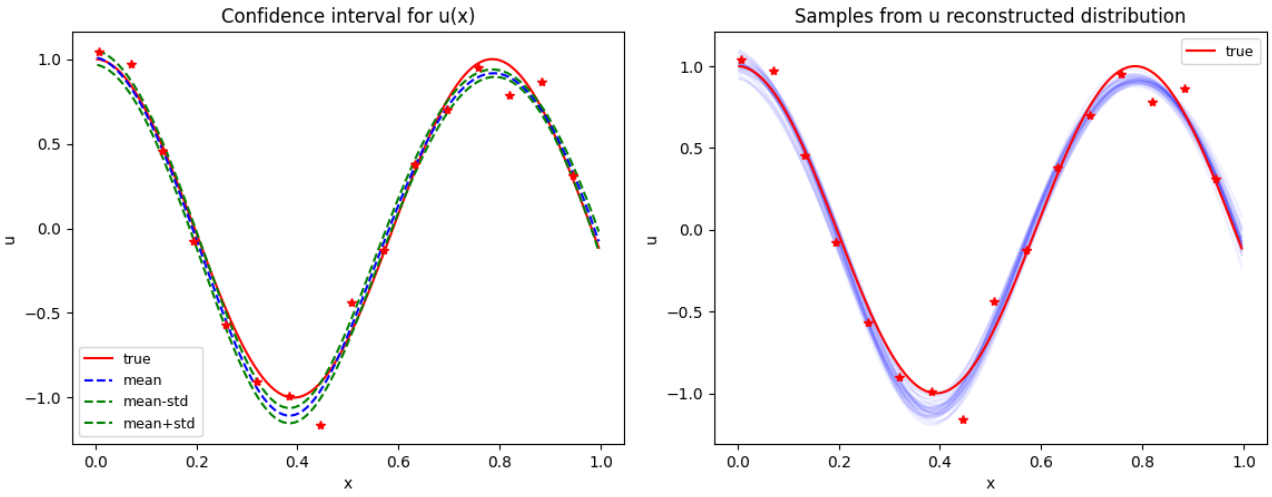
6.1.2. HMC

We then reconstructed the function with the first Bayesian model, with the HMC training method, imposing these options.

Parameter	Value
N	200
B	50
S	5
L	20
dt	10^{-3}
η	1

This parameter set was obtain through fine-tuning, which was not a trivial task given the amount of options and the co-implications behind them. This challenge became even more difficult when dealing with physics-informed tasks, as we will present in subsection 6.4.

On the other hand, HMC has a computational demand of medium level, if compared with the other algorithms implemented, and for this reason tuning to reach function reconstruction below the noise level was possible.



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 20: Function regression with HMC on $u(x) = \cos(8x)$

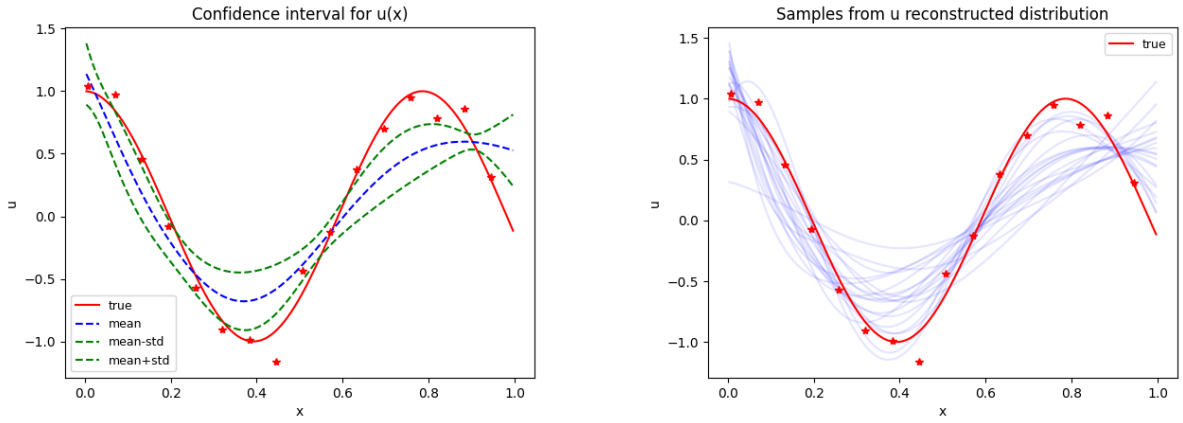
6.1.3. SVGD

For the SVGD case, the method-specific options were those one. They enabled to obtain reasonable results, but the high computational demand of the algorithm represented the main obstacle for parameter tuning.

Parameter	Value
E	2000
B	100
N	20
h	1000
ε	10^{-5}

What affects significantly the computational overload requested by this algorithm is the number of particles N, and the limitation on their number represents a weaken factor for the quality of UQ. On the other hand, the heuristic recipe to improve the prediction is this time simpler than with HMC, because it is necessary to increase number of epochs E and of particles N.

To reduce the computational overload, a possibility that could be exploited in further extension of the project is the parallelization of the algorithm, that in the current implementation has already a syntax inspired to parallel programming.



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

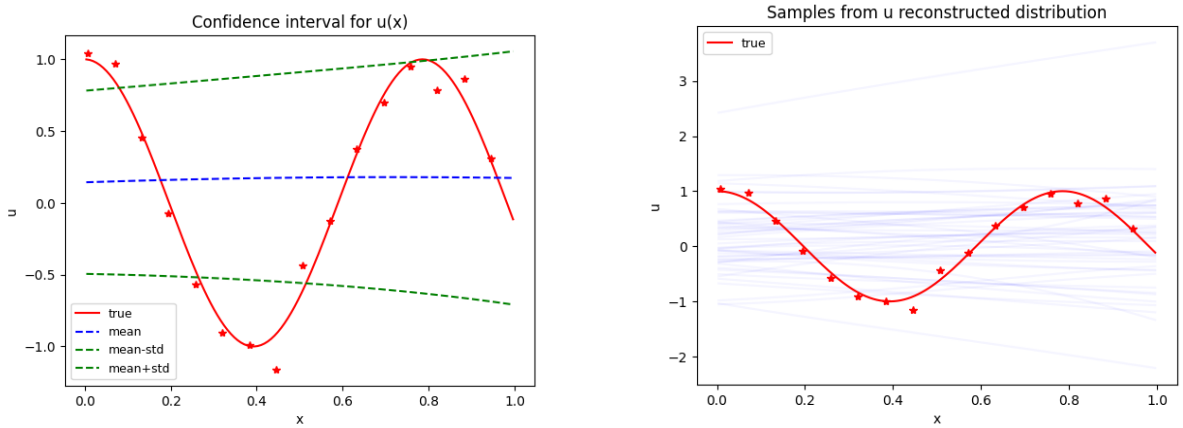
Figure 21: Function regression with SVGD on $u(x) = \cos(8x)$

6.1.4. VI

This algorithm was the less time-demanding for training among the Bayesian choices, and, as remarked in subsection 3.2, it enables to get cheaply as many function samples as desired, improving therefore the UQ.

Parameter	Value
N	2000
M	50
α	10^{-5}

As a downside, we obtained a poor performance in this case as it happened to VI cases in the reference [11]. This is due to the underlying absence of correlation in the distribution of the reconstructed network parameters, due to the choice of the multivariate gaussian with diagonal covariance matrix as proposed in [2].



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 22: Function regression with VI on $u(x) = \cos(8x)$

6.2. Damped Harmonic Oscillator Problem

In this section, we propose two models to tackle the differential problem of the Damped Harmonic Oscillator, presented in subsection 5.5. The aim of this application is mainly to highlight the model improvement that arises from the introduction of the physical information, which was completely absent in subsection 6.1.

This contribution becomes indeed evident when available data are not equidistributed, but restricted to some domain regions, as it can happen when trying to make a prediction for the evolution of a quantity over time, having at hand only data referred to the past time window (as done in [7] for COVID-19 previsions).

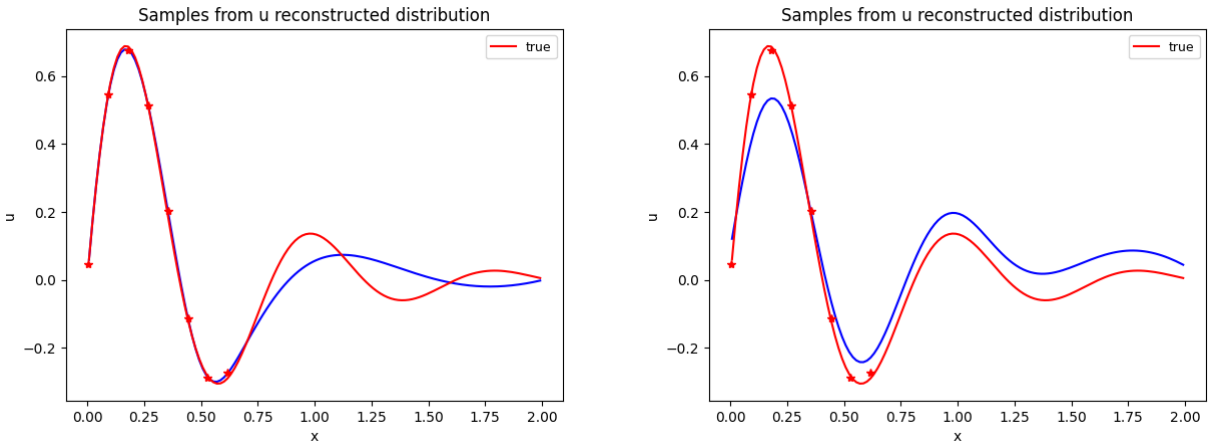
In this section, we present a comparison between a NN and a PINN that rely on the Adam deterministic training. However, since especially in the forecasting setting it makes sense to produce confidence intervals for the prediction, in subsection 6.4 we will present a tuned B-PINN for this problem.

The Damped Harmonic Oscillator under analysis has $\delta = 2$ and $\omega = 8$; moreover, we suppose to have $N_{fit} = 8$ noisy measurements ($\sigma_u = 0.01$), situated in the time subinterval $[0, 0.7]$, which mimic the initial evolution of the quantity under analysis.

6.2.1. NN vs PINN

In Figure 23a, we show the reconstruction of the solution provided by a NN that relies only on measurements: in the time subinterval $(0.7, 2]$, the profile is totally missed, and retrieved only at the final time thanks to the imposition of the boundary conditions.

We therefore trained a PINN to reconstruct the profile of the oscillations thanks to the presence of the physical law within the loss function. The introduction of this information needed to be carefully added because of the magnitude of the pde residual component of the loss.
Therefore, we balanced it with the choice of a higher uncertainty on the PDE, setting $\sigma_r = 10$.



(a) NN solution

(b) PINN solution

Figure 23: Damped Harmonic Oscillator Problem

6.3. Multi-dimensional domain

The proposed implementation is able to consider even multi-dimensional cases, because all the operations (even the differential ones) are coded in such a way that they can handle non-scalar quantities.

While the code executed is exactly the same for any arbitrary dimension up to the error and UQ, the only difference consists in the visualization option, for which we chose heatmaps for the 2D case.

6.3.1. Adam 2D cosine

Figure 24 and Figure 25 show a comparison between exact and mean predicted PINN output when solving the 2D Laplace problem on the unit square $[0, 1] \times [0, 1]$.
The functions under analysis are:

$$u(x, y) = \cos(8x) + \cos(8y) \qquad f(x, y) = 64\cos(8x) + 64\cos(8y)$$

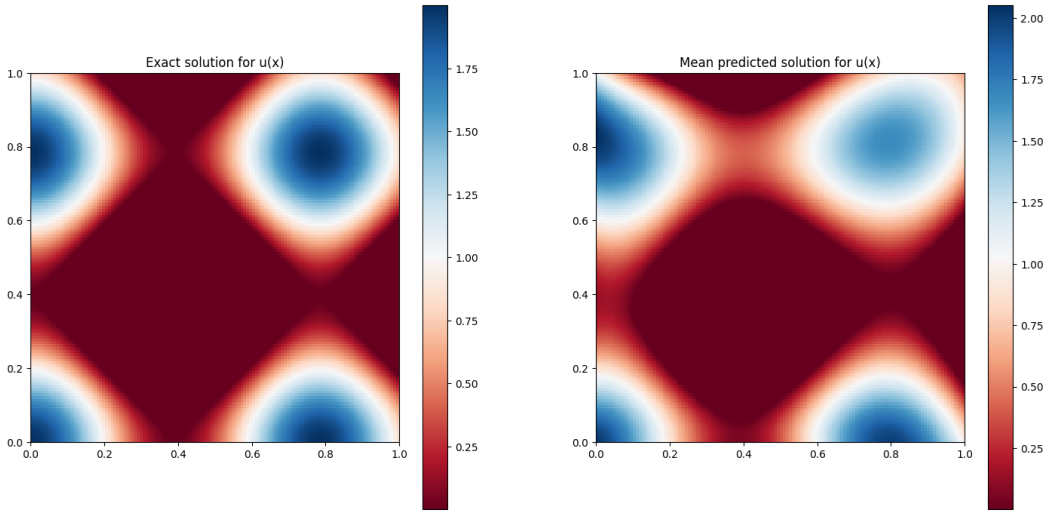


Figure 24: Reconstruction of $u(x, y)$ with an Adam-trained PINN

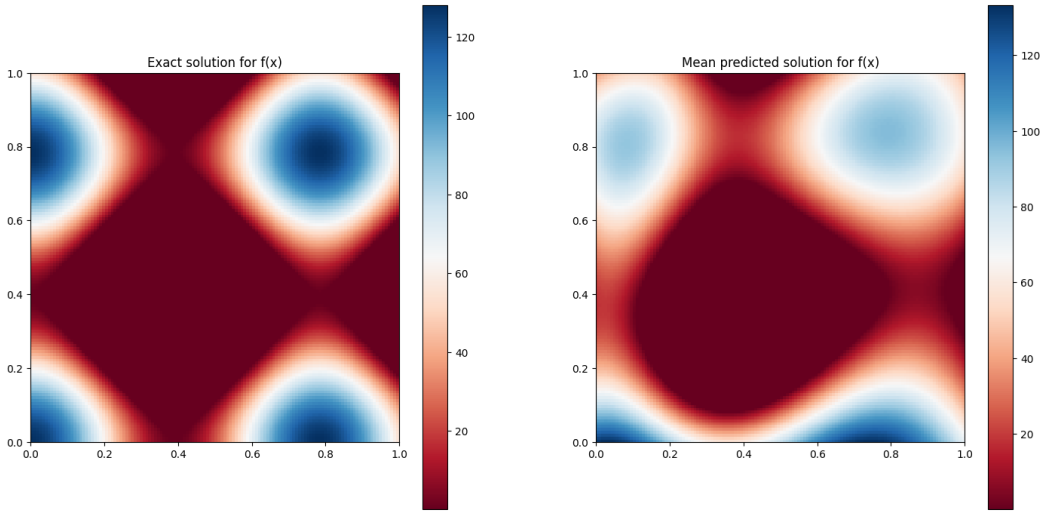


Figure 25: Reconstruction of $f(x, y)$ with an Adam-trained PINN

6.4. HMC Showcase

In this section, we deepen into some results obtained with the HMC training algorithm, either on function interpolation and on differential problems involving the Poisson/Laplace and Damped Harmonic Oscillator equations.

We concentrated the tuning effort on this algorithm because of the interesting parameters co-implications in their choice, and also because the medium computational time made it possible to run several trainings on our CPUs.

6.4.1. Regression Problem - sine

In this experiment, we implemented a BNN for the reconstruction of the function $u(x) = \sin^3(6x)$ in the domain $[-1, 1]$, relying on $N_{fit} = 16$ noisy measurements ($\sigma = 0.1$).

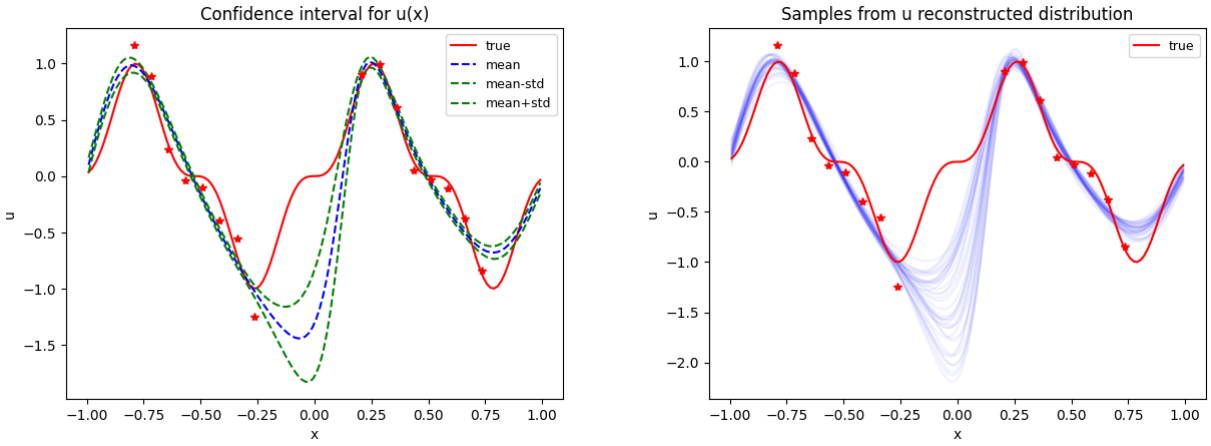
This test case shows a function with a more tangled graph with respect to the cosine example proposed in subsection 6.1, whose details we will try to teach to the network in a physics-informed framework as the one proposed at the end of this section.

We generated our data localizing them in two subdomains with the feature described in subsection 5.3, assuming to have measurements available only in the set $[-0.75, -0.25] \cup [0.25, 0.75]$.

When analyzing the results in Figure 26, apart from remarking the evidence of the missing information in the region $[-0.25, 0.25]$ that will be overcome with the introduction of the PDE residual in the loss, we focus on what happens to UQ in the region where data are not available.

Indeed, we can see that the confidence interval is reasonably broader, because in the areas where the constraint of data is not available, the network is more flexible and leaves the door open to bigger uncertainty in the prediction.

Parameter	Value
N	500
B	100
S	10
L	20
dt	10^{-3}
η	1



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 26: Function regression with HMC on $u(x) = \sin^3(6x)$

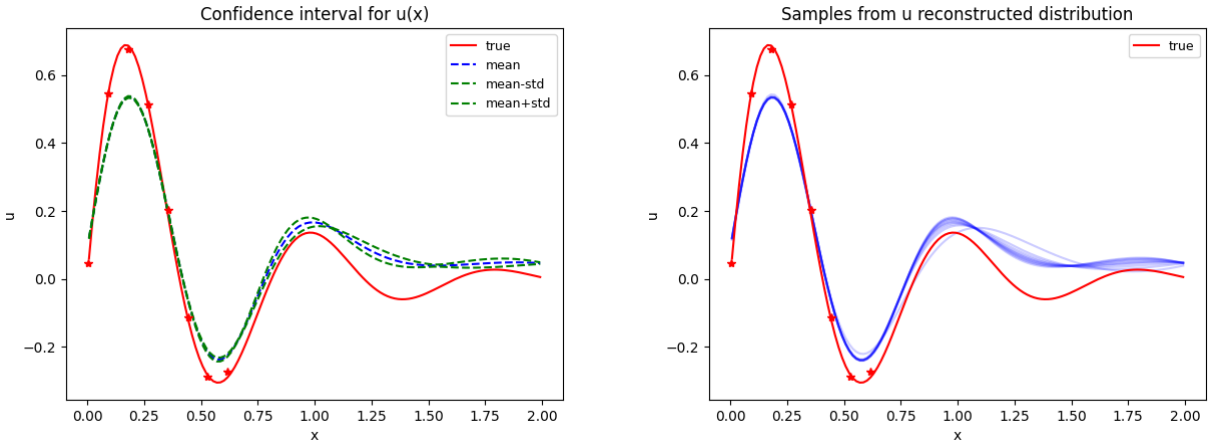
6.4.2. Oscillator Problem

This experiment shows the HMC counterpart of the same oscillator problem presented in subsection 6.2, by adding a third element of comparison: after NN and PINN, we here propose a BPINN.

While on the physics-informed side the PINN and the BPINN are exactly in parity, what the BPINN adds is a confidence interval for the prediction.

To reproduce the test case in Figure 27, we ran $N = 3000$ Adam epochs, then let HMC start the options aside. For the uncertainty on the PDE residual, we employed the same noise level of the PINN $\sigma_r = 10$.

Parameter	Value
N	100
B	0
S	10
L	200
dt	5×10^{-5}
η	0.5



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 27: Damped Oscillator Problem with HMC

6.4.3. Poisson/Laplace Problem - cosine

In these last two sections, we propose the solution of the Laplace 1D problem with two different datasets. The first one we present involves two cosine functions in the domain $[0, 1]$:

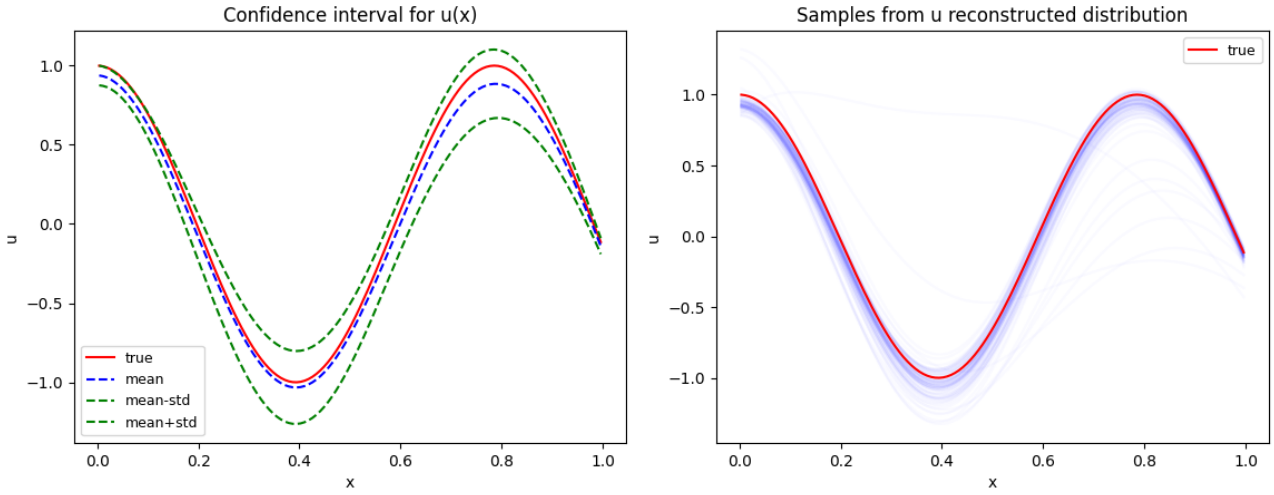
$$u(x) = \cos(8x) \qquad f(x) = 64\cos(8x)$$

For what concerns the availability of measurements, we assume to have $N_{fit} = 16$ noisy data ($\sigma_f = 0.1$) on the forcing term f and only the boundary conditions for the solution u. Indeed, we want to reconstruct the solution using the PDE without relying on any measurements - apart from boundary conditions.

In this case, the magnitude of the physical loss was comparable to the one of the fitting loss. Therefore, we set its uncertainty to the same value of the noise level, resulting in $\sigma_r = 0.1$.

In Figure 28 and Figure 29 we show the reconstruction of the functions involved in the problem: coherently with the intuition, confidence intervals are wider for the solution, for which the constraint of measurements lacks, and samples of the solution create an envelope around the ground truth, while samples of the forcing term are closer to each other.

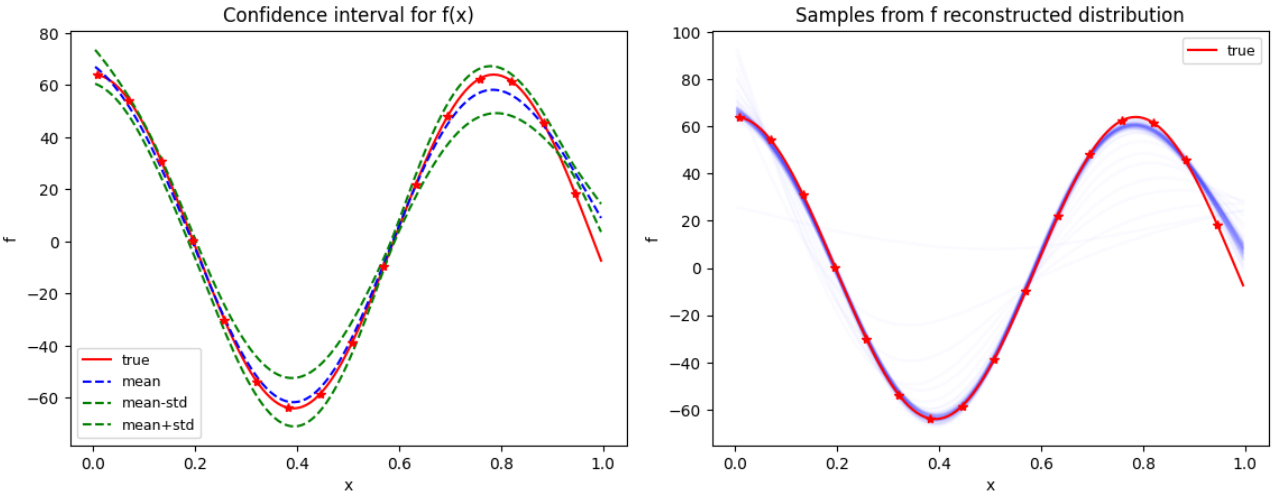
Parameter	Value
N	1000
B	70
S	10
L	20
dt	10^{-3}
η	0.5



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 28: Poisson/Laplace Cosine Problem with HMC - solution



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 29: Poisson/Laplace Cosine Problem with HMC - parametric field

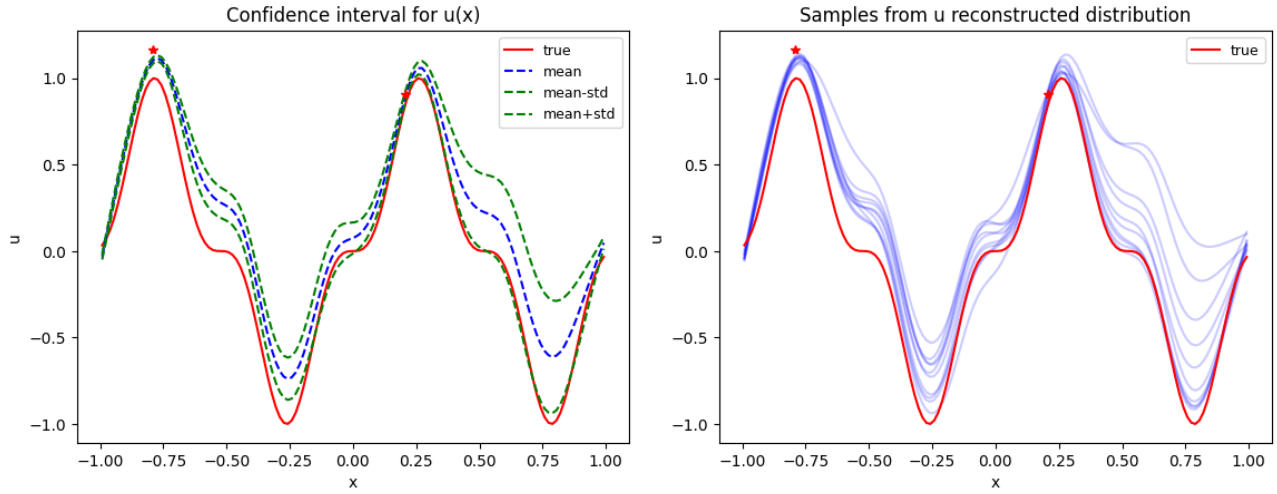
6.4.4. Poisson/Laplace Problem - sine

Finally, we proposed the same problem on a different dataset: we considered the Poisson Problem with the function $u(x) = \sin^3(6x)$ as solution in the domain $[-1, 1]$. The forcing term corresponding to this solution is $f(x) = 108\sin(6x)(-2\cos(6x)^2 + \sin(6x)^2)$, and we assumed to have $N_{fit} = 32$ noisy measurements on that, with $\sigma_f = 0.1$.
The higher number of fitting point required to have satisfying results is linked to the extremely oscillatory behavior of the function f, which was difficult to learn for the network.

Another consequence of the difficulty derived by the tangled function profiles was the necessity to include a deterministic Adam pre-training before the start of the bayesian algorithm, to make HMC start after a fairly good reconstruction of the oscillations of the function f.

We ran $N = 11000$ epochs of Adam with the optimal parameters ([5]) and then, HMC with the training showed here. After performing a test case without measurements on the solution u, we noticed that the BPINN was able to reconstruct the function profile up to an additive constant. Therefore, we propose in Figure 30 a solution with $N_{fit} = 2$ noisy data also on the solution u, attempting to fix the level of the constant missed by the network.

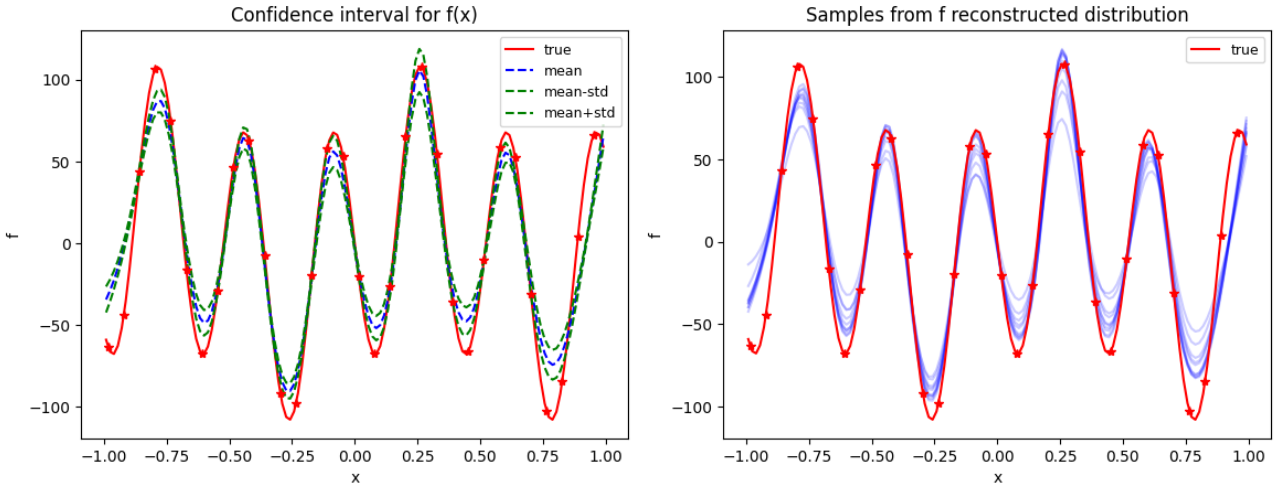
Parameter	Value
N	100
B	0
S	10
L	200
dt	10^{-4}
η	0.5



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 30: Poisson/Laplace Sine Problem with HMC - solution



(a) Prediction with confidence interval.

(b) Ensemble of all the samples.

Figure 31: Poisson/Laplace Sine Problem with HMC - solution