Supplementary Materials

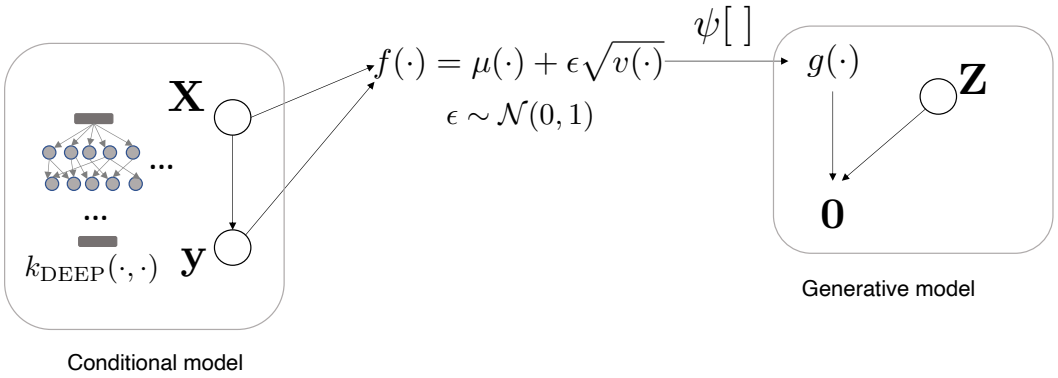
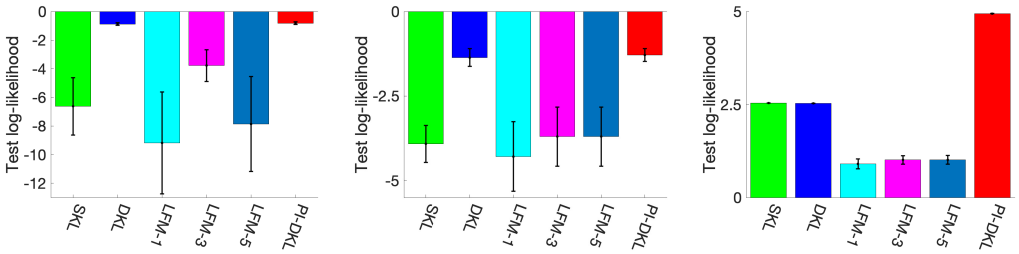


Figure 1: Graphical representation of Physics Informed Deep Kernel Learning (PI-DKL). The physics knowledge are encoded by a differential equation $\psi[f(\mathbf{x})] = g(\mathbf{x})$. The conditional model is the deep-kernel GP and the generative model is equivalent to placing a GP prior over $g(\cdot)$.

1 Test Log-likelihood on Real-World Datasets

In Fig. 2 and 3, we report the test log-likelihood (LL) of all the methods in the real-world applications in Section 6.2 of the main paper. Note that since test LLs are negative (smaller than zero) in most datasets, the corresponding bar plots are shown inverted for a convenient comparison. As we can see, our method (PI-DKL) consistently outperforms all the competing methods, and in many cases by a large margin. DKL always obtains test LLs larger than or comparable to SKL except that in Fig. 3 a, DKL is lightly worse. It demonstrates the advantage of more expressive kernels. PI-DKL further improves upon DKL in all the cases, showing that the physics knowledge are effectively exploited and indeed help with the prediction. Especially, in Fig. 3a, while DKL obtains slightly smaller test LLs than SKL, after PI-DKL regularizes the same deep kernel with physics, the test LLs are greatly improved. Note that, similar to nRMSE results, we can see LFM improves upon SKL in some cases, *e.g.,* LFM-3 in Fig. 2 a and b, but in other cases are even worse, *e.g.,* in Fig. 2 c. This might because the rigid incorporation (hard-coding) of the physics in LFM can even hurt the performance when there is a significant mismatch to the actual data. For example, a first-order ODE might be too simple to describe the motion data in Fig. 2 c. Overall, the test LL results are consistent with nRMSEs shown in the main paper.

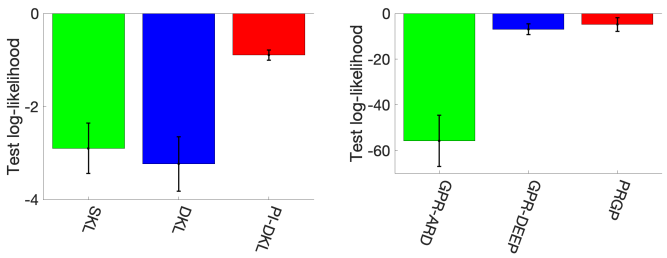


(a) Copper (b) Cadmium (c) Motion

Figure 2: Test log-likelihood (LL) in Swiss Jura (a, b) and joint angle prediction in motion capture (c). The results are averaged over 5 runs.

2 Marginal Distribution of g

We have $\mathbf{h} = [h(\mathbf{z}_1, \epsilon), \ldots, h(\mathbf{z}_m, \epsilon)]$, where $h(\cdot, \epsilon) = \psi[\mu(\cdot) + \epsilon\sqrt{v(\cdot)}]$ (see (6) of the main paper). To obtain each element $\widetilde{g}_j$ in $\mathbf{g}$, we can first sample the Gaussian random noise, $\epsilon \sim \mathcal{N}(\epsilon|0,1)$, and sample $\widetilde{g}_j \sim \delta\left(\widetilde{g}_j - h(\mathbf{z}_j, \epsilon)\right)$ (see (7) of the main paper). However, we can also consider the marginal distribution each $\widetilde{g}_j$. Since $\widetilde{g}_j$ is a transformation of Gaussian noise ϵ, $\widetilde{g}_j = \alpha_j(\epsilon)$ where



(a) Copper (b) Cadmium

Figure 3: Test log-likelihood (LL) for PM2.5 and traffic flow datasets. The results are averaged over 5 runs.

$\alpha_j(\cdot) = h(\mathbf{z}_j, \cdot)$. The marginal distribution of $\widetilde{g}_j$ is

$$p(\widetilde{g}_j|\mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\alpha_j^{-1}(\widetilde{g}_j)|0, 1\right)|\frac{\mathrm{d}\epsilon}{\mathrm{d}\widetilde{g}_j}|. \tag{1}$$

Although conceptually available, the marginal distribution is tricky to compute — the transformation $\alpha_j(\cdot)$ couples complex differential operators in ψ and nonlinear functions $\mu(\cdot)$ and $v(\cdot)$. The inverse $\alpha_j^{-1}(\cdot)$ is very complicated and likely to have no closed-forms. The marginal joint distribution for $\mathbf{g}$ will be even more difficult to compute. Therefore, we choose to explicitly sample ϵ and then obtain the sample for $\mathbf{g}$.