

Fig.1. 示例方程使用常规神经网络(NN)时, 通过在整个时间域内均匀取样26个数据点($N_{\text{data}} = 26$), 得到的求解情况(左图)。使用物理信息神经网络(PINN), 用均匀取样的50个配置点($N_c = 50$)来计算常微分方程损失函数LF, 数据点 $N_{\text{data}} = 26$, 同样取自整个区间。评估物理损失函数的时间值(配置点)用t轴上的小红色圆点表示。

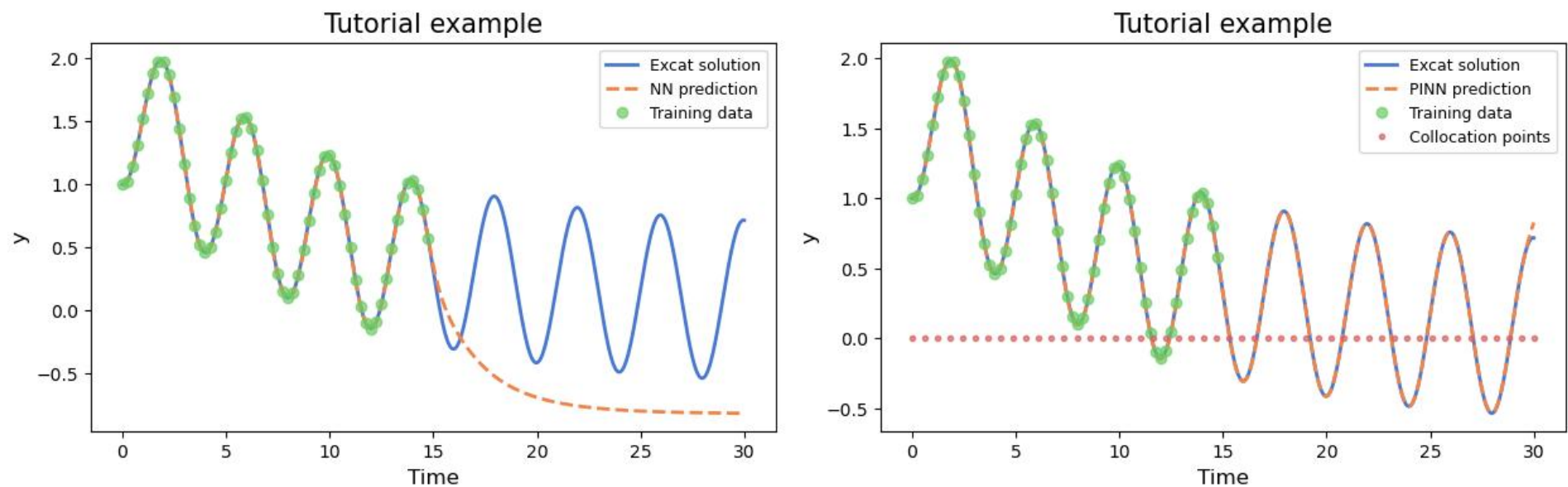


Fig.2. 示例方程使用常规神经网络(NN), 在左侧子区间内取60个数据点($N_{\text{data}} = 60$), 得到的求解情况(左图)。使用物理信息神经网络(PINN), 同样在左侧子区间内取得60个数据点($N_{\text{data}} = 60$), 用均匀取样50个配置点($N_c = 50$)得到的求解情况(右图)。

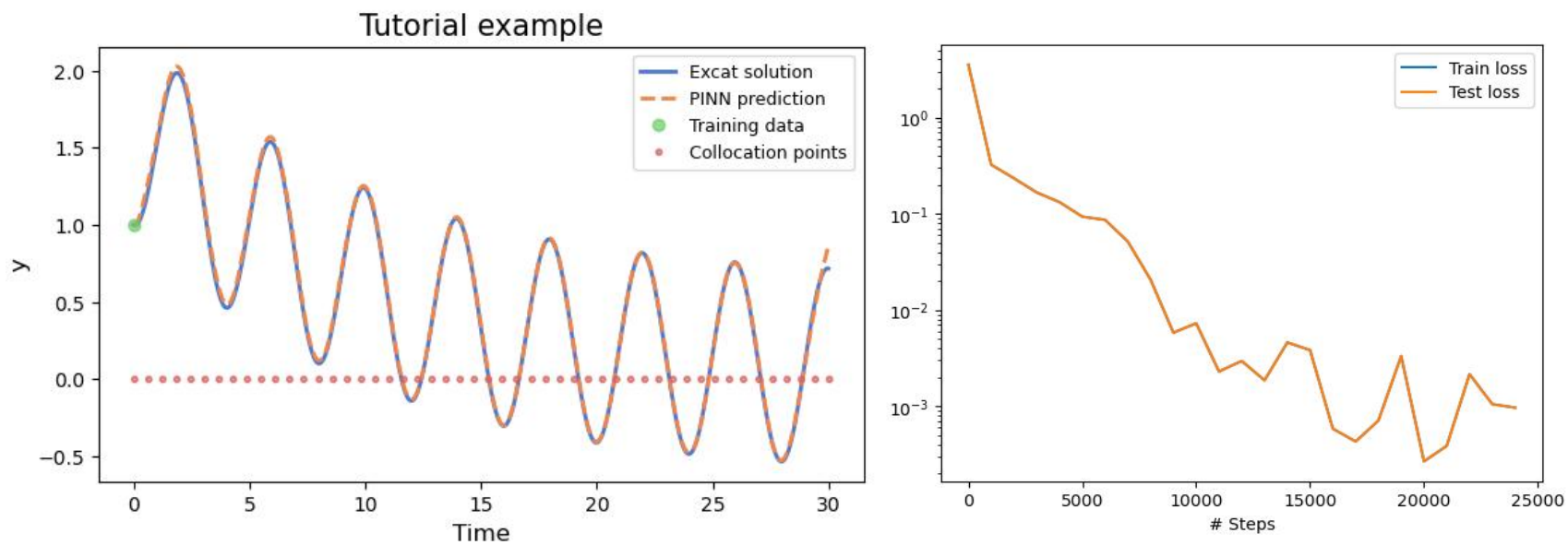


Fig.3. 示例方程使用物理信息神经网络(PINN)得到的求解情况，其中 $N_{\text{data}} = 1$ ($t = 0$ 时施加的初始条件)， $N_c = 50$ 个配置点(左图)。相应的损失函数历史记录(右图)。

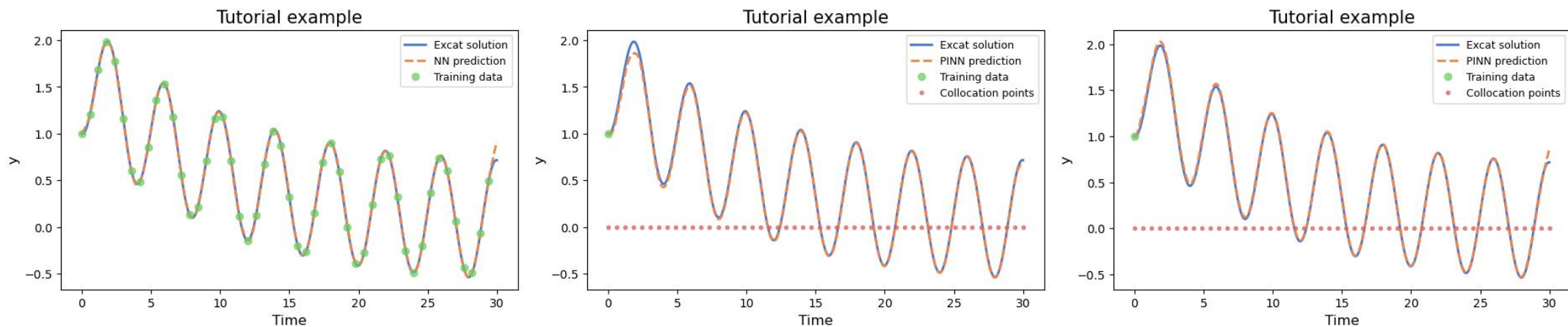


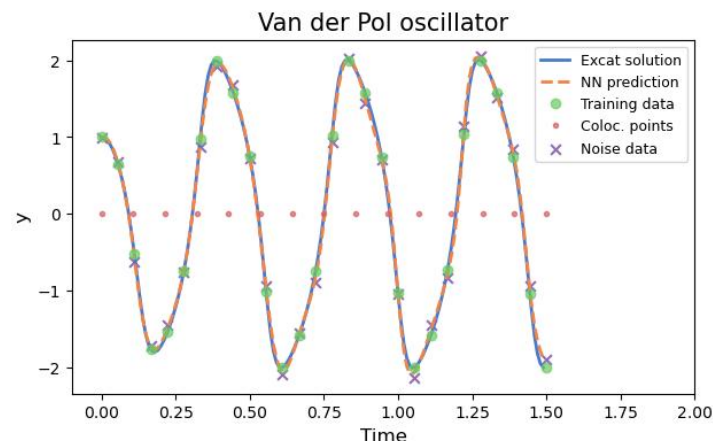
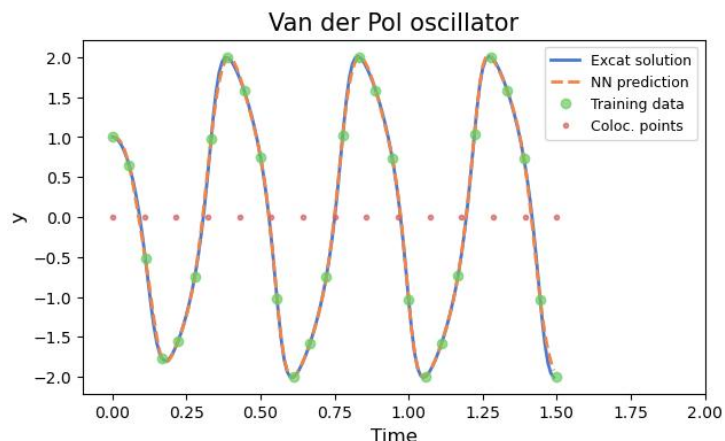
Fig.4. 示例方程使用常规神经网络(NN)得到的最优求解情况，其中 $N_{\text{data}} = 50$ (最少训练点)，训练时间为10.53秒(左图)。使用Pytorch实现物理信息神经网络(PINN)得到的最优求解情况，其中 $N_{\text{data}}=1$ ($t = 0$ 时施加的初始条件)， $N_c = 50$ 个配置点，训练时间为17.30秒(中间图)。使用DeepXDE实现物理信息神经网络(PINN)得到的最优求解情况，其中 $N_{\text{data}}=1$ ($t = 0$ 时施加的初始条件)， $N_c = 50$ 个配置点，训练时间为8.82秒(右图)。

Pytorch实现NN
训练时间：10.53s
50个训练点

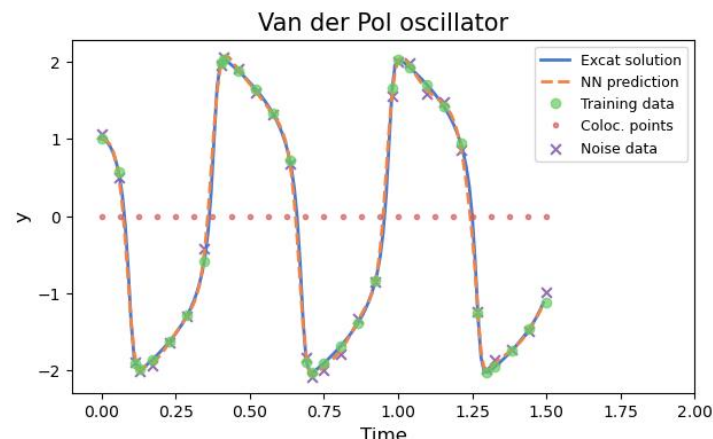
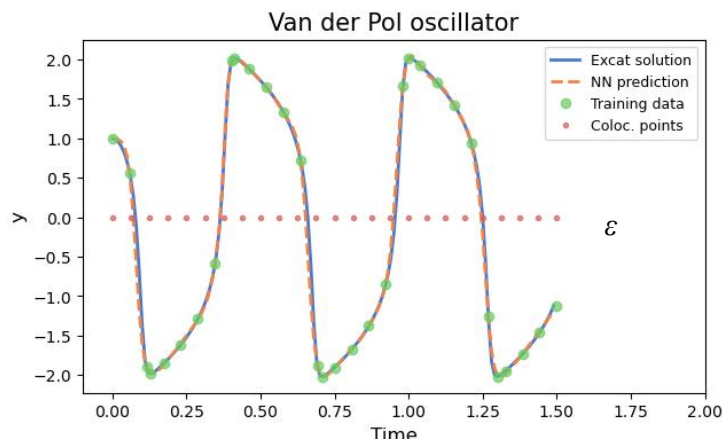
Pytorch实现PINN
训练时间：17.30s
1个训练点
50个配置点

DeepXDE实现PINN
训练时间：8.82s
1个训练点
50个配置点

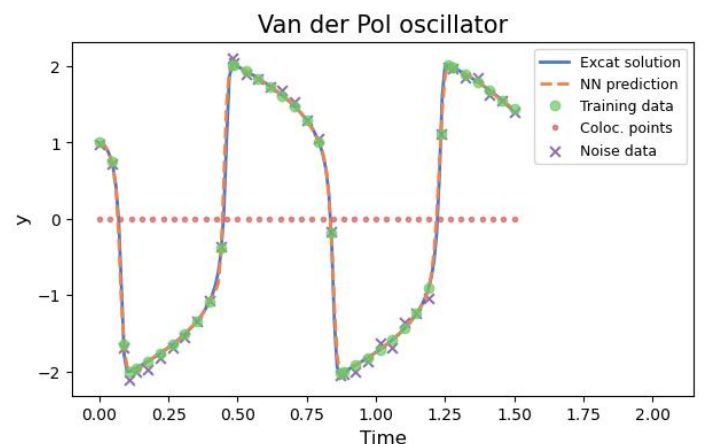
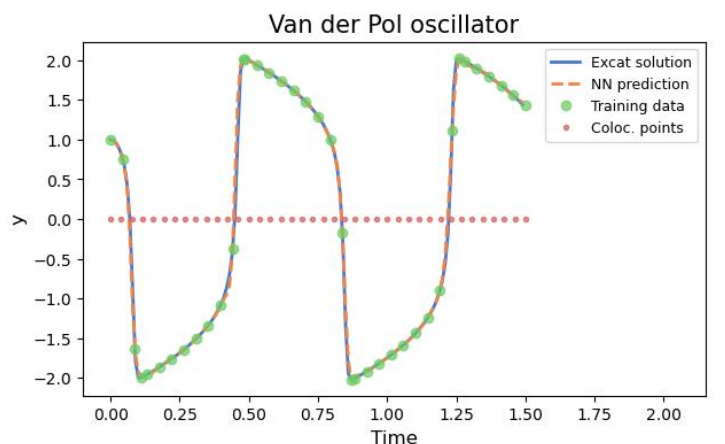
Fig.5. 在不同参数下VDP方程的PINN方法求解情况及容忍最大噪声的能力。在顶图中，系数为1， $N_{\text{data}}=28$ ， $N_c=15$ ，Noise=0.1。在中图中，系数为3， $N_{\text{data}}=32$ ， $N_c=25$ ，Noise=0.08。在底图中，系数为5， $N_{\text{data}}=38$ ， $N_c=40$ ，Noise=0.05。



系数=1
 $N_{\text{data}}=28$
 $N_c=15$
 Noise=0.1



系数=3
 $N_{\text{data}}=32$
 $N_c=25$
 Noise=0.08



系数=5
 $N_{\text{data}}=38$
 $N_c=40$
 Noise=0.05

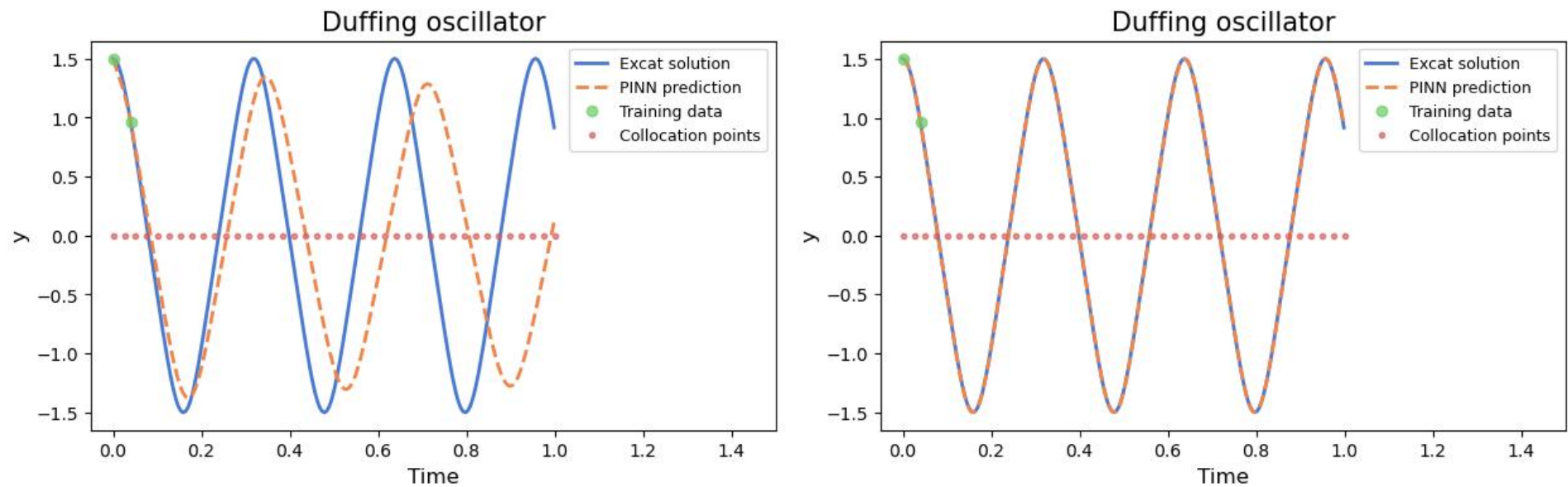


Fig.6. 使用PINN方法验证能量守恒对杜芬振荡器求解情况的影响，其训练轮次为36000， $N_{\text{data}}=2$ ， $N_c=40$ 。无能量守恒求解情况(左图)，添加能量守恒求解情况(右图)。能量守恒为 $E = \frac{1}{2} \left(\frac{dy}{dt} \right)^2 + \frac{1}{4} w_0^2 y^4$ (由初始条件确定的常数相加)，该图的常数值为 $y_0=1.5$ ， $w_0=15.5$ 。