Small Data — Big Data

Maziar Raissi

Physics Based — Physics Free

Forward-Backward Stochastic Neural Networks

Deep Learning of High-dimensional Partial Differential Equations

$$
\begin{aligned}
Y_t &= u(t, X_t)\\
Z_t &= Du(t, X_t)
\end{aligned}
$$

$u(t,x)$ → **Deep Neural Network**

$Du(t,x)$ → **Automatic Differentiation**

$$
\begin{aligned}
dX_t &= \mu(t, X_t, Y_t, Z_t)dt + \sigma(t, X_t, Y_t)dW_t\\
X_0 &= \xi\\
dY_t &= \varphi(t, X_t, Y_t, Z_t)dt + Z_t'\sigma(t, X_t, Y_t)dW_t\\
Y_T &= g(X_T)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \varphi(t,x,u,Du) - \mu(t,x,u,Du)'Du - \frac{1}{2}\mathrm{Tr}[\sigma(t,x,u)\sigma(t,x,u)'D^2u]\\
u(T,x) &= g(x)
\end{aligned}
$$

→ **Quasi-Linear Partial Differential Equation**

↓

Forward-Backward Stochastic Differential Equation

$$
\begin{aligned}
X^{n+1} &\approx X^n + \mu(t^n, X^n, Y^n, Z^n)\Delta t^n + \sigma(t^n, X^n, Y^n)\Delta W^n\\
Y^{n+1} &\approx Y^n + \varphi(t^n, X^n, Y^n, Z^n)\Delta t^n + (Z^n)'\sigma(t^n, X^n, Y^n)\Delta W^n\\
\Delta W^n &\sim \mathcal{N}(0, \Delta t^n)
\end{aligned}
$$

→ **Euler-Maruyama Scheme**

$$
\sum_{m=1}^{M}\sum_{n=0}^{N-1} |Y_m^{n+1} - Y_m^n - \varphi(t^n, X_m^n, Y_m^n, Z_m^n)\Delta t^n - (Z_m^n)'\sigma(t^n, X_m^n, Y_m^n)\Delta W_m^n|^2 + \sum_{m=1}^{M} |Y_m^N - g(X_m^N)|^2
$$

→ **Loss Function**

$$
Y_m^n = u(t^n, X_m^n) \qquad Z_m^n = Du(t^n, X_m^n) \qquad X_m^{n+1} = X_m^n + \mu(t^n, X_m^n, Y_m^n, Z_m^n)\Delta t^n + \sigma(t_m^n, X_m^n, Y_m^n)\Delta W_m^n \qquad X_m^0 = \xi
$$

Raissi, Maziar. "Forward-Backward Stochastic Neural Networks: Deep Learning of High-dimensional Partial Differential Equations." *arXiv preprint arXiv:1804.07010* (2018).

Small Data | Big Data

Maziar Raissi

Physics Based | Physics Free

Forward-Backward Stochastic Neural Networks

Deep Learning of High-dimensional Partial Differential Equations

$$u_t = -\frac{1}{2}\mathrm{Tr}[0.16\ \mathrm{diag}(X_t^2)D^2u] + 0.05(u - (Du)'x)$$
$$u(1,x) = \|x\|^2$$

↓

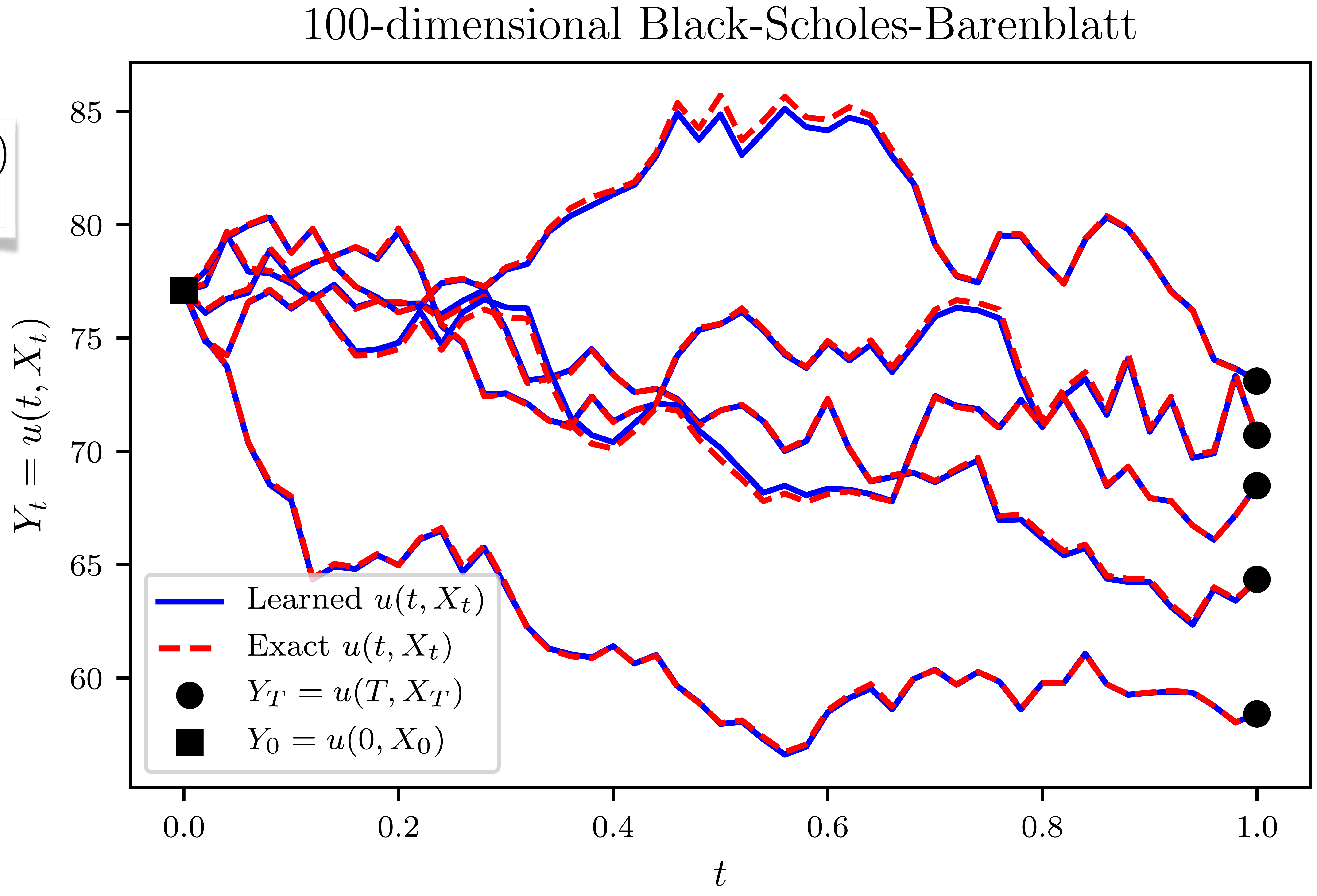
Black-Scholes-Barenblatt Equation in 100D

$$dX_t = 0.4\ \mathrm{diag}(X_t)dW_t, \quad t \in [0,1]$$
$$X_0 = (1, 0.5, \ldots, 1, 0.5)$$
$$dY_t = 0.05(Y_t - Z_t'X_t)dt + 0.4\ Z_t'\mathrm{diag}(X_t)dW_t$$
$$Y_1 = \|X_1\|^2$$

↓

Forward-Backward Stochastic Differential Equation

$$u(t,x) = \exp\left(0.21(1-t)\right)\|x\|^2$$ → **Exact Solution**



Raissi, Maziar. "Forward-Backward Stochastic Neural Networks: Deep Learning of High-dimensional Partial Differential Equations." *arXiv preprint arXiv:1804.07010* (2018).

Small Data | Big Data

Maziar Raissi

Physics Based | Physics Free

Forward-Backward Stochastic Neural Networks

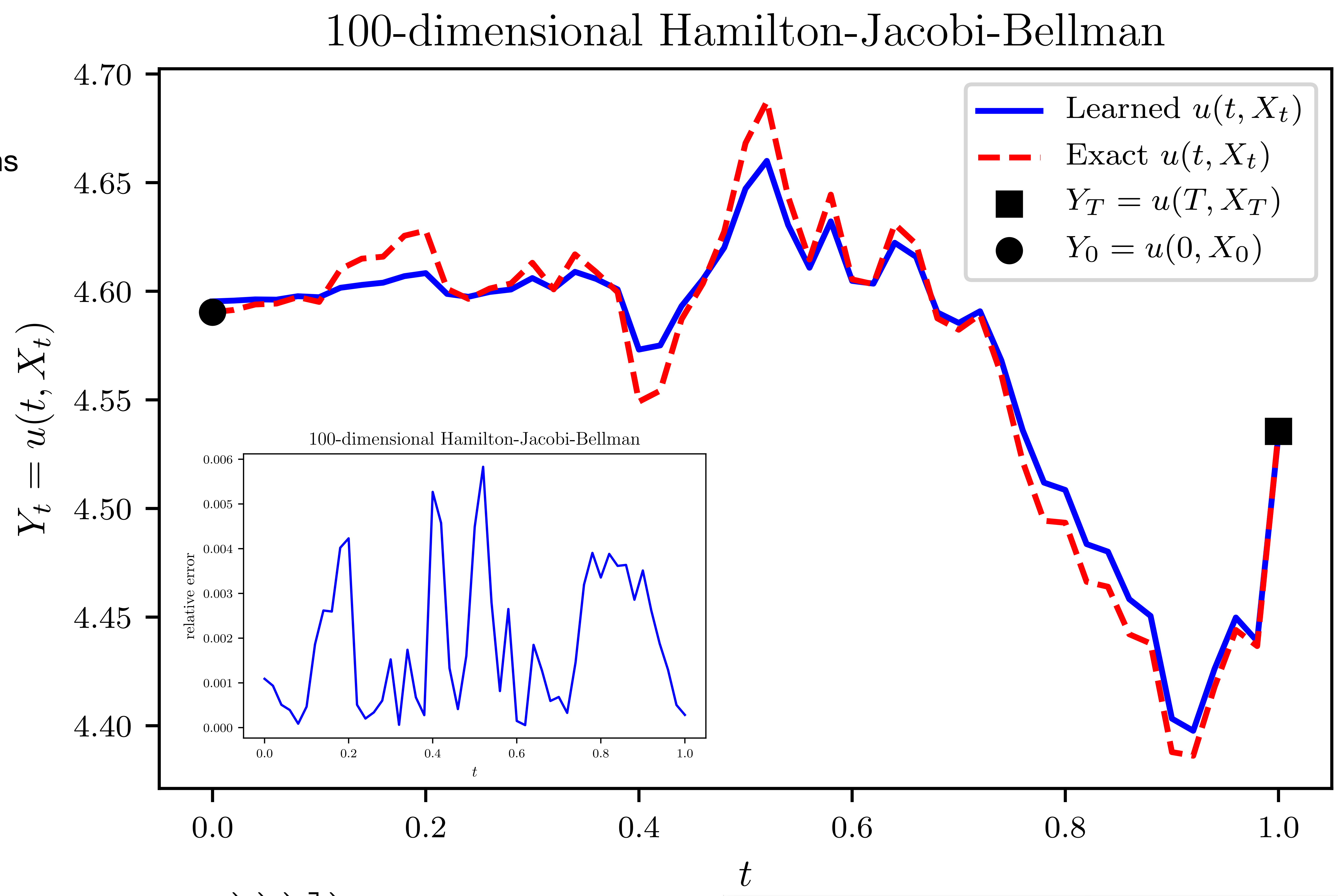
Deep Learning of High-dimensional Partial Differential Equations

$$u_t = -\mathrm{Tr}[D^2 u] + \|Du\|^2$$
$$u(1,x) = \ln\left(0.5\left(1 + \|x\|^2\right)\right)$$

Hamilton-Jacobi-Bellman Equation in 100D

$$dX_t = \sqrt{2} dW_t, \quad t \in [0,1]$$
$$X_0 = (0,0,\ldots,0)$$
$$dY_t = \|Z_t\|^2 dt + \sqrt{2} Z_t' dW_t, \quad t \in [0,1)$$
$$Y_1 = \ln(0.5(1 + \|X_1\|^2))$$

Forward-Backward Stochastic Differential Equation



$$u(t,x) = -\ln\left(\mathbb{E}\left[\exp\left(-\ln\left(0.5\left(1 + \|(x + \sqrt{2} W_{T-t})\|^2\right)\right)\right)\right]\right)$$ → **Exact Solution**

https://maziarraissi.github.io/FBSNNs/

Raissi, Maziar. "Forward-Backward Stochastic Neural Networks: Deep Learning of High-dimensional Partial Differential Equations." *arXiv preprint arXiv:1804.07010* (2018).