Enhancing FBSNN Performance for HJB Equations: Two Methodologies and a Failure Analysis

Su Rin Lee
Department of Artificial Intelligence, Chung-Ang University

***Abstract*—This study addresses a key limitation of the Deep Forward-Backward Stochastic Neural Network (Deep FBSNN) method when solving high-dimensional Partial Differential Equations (PDEs), specifically the 100-dimensional Hamilton-Jacobi-Bellman (HJB) equation. While the FBSNN accurately predicts the solution at the boundary points (Y_0, Y_T), a clear prediction discrepancy was observed within the internal time trajectory $t \in (0, T)$. Two modification strategies—a structural enhancement of the neural network (e.g., FourierMultiDeepONet) and a dynamic loss function re-weighting technique—were attempted to improve trajectory accuracy but failed to yield significant improvements. This failure was primarily attributed to the inability of architectural enhancements to introduce the fundamental change in function representation necessary for handling the strong nonlinearity, and the instability caused by the dynamic shifting of the input data due to the simulation of Stochastic Differential Equation (SDE) paths. This failure underscores a structural limitation: the existing FBSNN framework has an inherent bias that causes a decrease in accuracy despite loss minimization. The analysis demonstrates that the FBSNN framework, due to its stochastic and path-dependent nature, requires a more fundamental, non-local approach to error distribution than effective static grid-based loss balancing. This finding highlights the strong need for subsequent research into core structural changes for robust FBSNN performance in high-dimensional problems.**

I. Introduction

High-dimensional Partial Differential Equations (PDEs) are central to modeling complex phenomena across various scientific and engineering domains, including quantitative finance, physics, and optimal control theory, such as the Hamilton-Jacobi-Bellman (HJB) equation. However, classical grid-based numerical methods for solving these equations suffer from the Curse of Dimensionality, as the computational cost and resource requirements grow exponentially with the number of variables, rendering them impractical for dimensions exceeding three or four.

The rise of deep learning has introduced innovative solutions to circumvent this long-standing challenge. Specifically, the Deep Forward-Backward Stochastic Neural Network (Deep FBSNN) method leverages the well-known connection between high-dimensional PDEs and Forward-Backward Stochastic Differential Equations (FBSDEs), approximating the unknown solution $u(t, x)$ with a deep neural network [1]. The Deep FBSNN has proven highly effective in benchmark problems, successfully approximating solutions for quasi-linear equations like the Black-Scholes-Barenblatt (BSB) equation in high dimensions. However, when applied to the 100-dimensional Hamilton-Jacobi-Bellman (HJB) equation, a significant performance disparity emerges. While the method accurately matches the boundary conditions at the initial point $\mathbf{Y_0}$ and the terminal point $\mathbf{Y_T}$, a distinct prediction discrepancy is consistently observed within the internal time trajectory $\mathbf{t} \in (\mathbf{0}, \mathbf{T})$. This issue is critical, as the accuracy of the intermediate trajectory is essential for the real-time application and reliability of dynamic optimal control strategies governed by the HJB equation. The root of this disparity is hypothesized to lie in the strong non-linearity and the inclusion of an optimal control element inherent to the HJB equation, which complicates the solution landscape compared to the BSB equation.

This study aims to address this internal trajectory prediction error in the Deep FBSNN framework applied to the HJB equation, thereby enhancing its global accuracy and robustness. Diverging from prior works that focused primarily on refining numerical schemes or theoretical analysis, our approach takes a perspective rooted in Deep Learning (AI) methodologies, targeting the model's structure and its training optimization mechanism. We implemented two distinct modification strategies. The first was a Neural Network Structural Change. Specifically, we adopted and modified the DeepONet framework [2] by implementing a FourierMultiDeepONet—a Multi-Head architecture—intended to increase the model's explicit dependence on, and sensitivity to, the time variable t (temporal dependency). The second was a Dynamic Loss Reweighting Technique named Causality Loss Balancing. This technique involved dynamically and cumulatively adjusting the weights of the intermediate discretization loss terms relative to the terminal loss term, aiming to force the network to better fit the solution trajectory across intermediate time steps. The concept of introducing causality or temporal structure into neural network training has recently gained traction in related fields [3].

Unfortunately, both the structural modification and the optimization technique failed to yield significant improvements in the internal trajectory accuracy of the HJB solution. This failure was attributed to two primary factors, providing critical insights into the fundamental limitations of the Deep FBSNN framework. The first factor was a Structural and Technical Constraint Failure. The Multi-Head structure, upon closer inspection, amounted to little more than a simple concatenation of MLPs, failing to introduce a fundamental change in the model's capacity to represent the complex function $u(t, x)$. Furthermore, constraints related to hardware (GPU) and the technical feasibility of perfect multi-threading prevented the realization of any expected time efficiency gains. The second factor, which we identify as the Primary Cause, was the Causality Loss Balancing Failure. The failure of the loss balancing technique stems from the Deep FBSNN's core mechanism: the continuous, dynamic sampling of spatio-temporal coordinates by simulating new SDE paths at every training iteration. This constant shifting of the input data distribution prevents the effects of any static or semi-static loss balancing scheme from achieving stable and consistent efficacy. In addition, our analysis suggests the presence of a fundamental bias within the standard FBSNN loss function structure, which imposes an intrinsic limit on trajectory prediction accuracy even when the loss value is successfully minimized. This observation is corroborated by recent theoretical work that has identified and analyzed inherent biases within the FBSNN framework [4].

This study demonstrates a crucial structural limitation. While loss balancing methods may be effective in static, grid-based frameworks (like PINNs), the application of Causality Loss Balancing to the stochastic and path-dependent Deep FBSNN framework runs into fundamental structural barriers. Our failure analysis provides an invaluable insight, underscoring the necessity for future research to move beyond simple architectural or training adjustments and to focus instead on revising the core loss structure and the integration of the stochastic processes for robust trajectory prediction.

II. Related Work

A. High-Dimensional PDE Solvers: Deep FBSNN

The Deep FBSNN emerged as an innovative framework to solve the Curse of Dimensionality in high-dimensional PDEs by leveraging the connection between FBSDEs and deep learning. This approach, pioneered by works on Deep BSDEs, has proven successful for quasi-linear equations like the BSB. However, the method faces persistent challenges in accurately predicting the internal trajectory ($t \in (0,T)$) for the strongly nonlinear HJB equation, which remains a key open problem that needs to be resolved for reliable dynamic control applications. [1]

B. Architectural Innovation: DeepONet and Multi-Head Structures

To enhance the function approximation capability of the Deep FBSNN, our structural attempt was rooted in the DeepONet architecture [2], which excels in operator learning by separating input features into Branch and Trunk networks. This work specifically utilized a FourierMultiDeepONet, integrating a Multi-Head structure and Fourier Feature Mapping. This design was intended to improve the model's spatio-temporal dependency representation while providing potential for parameter efficiency and accelerated training through complex parallelization.

C. Causal Loss and Temporal Optimization

The challenge of optimizing time-dependent problems, where errors at early time steps affect predictions at later steps, is known in both traditional numerical methods and deep learning solvers like Physics-Informed Neural Networks (PINNs). The work "Respecting Causality Is All You Need for Training Physics-Informed Neural Networks" introduced a causal loss function designed to mitigate this issue [3]. This method involves applying a large weighting factor to the loss at earlier time steps. The goal is to force the network to prioritize learning the solution accurately across the entire trajectory, rather than allowing the loss at the final time T to dominate the optimization. Our implementation of Causality Loss Balancing was conceptually inspired by this principle, aiming to impose temporal constraints on the loss landscape to specifically improve the fidelity of the intermediate $t \in (0,T)$ solution path in the stochastic FBSNN setting.

III. Methodology

A. The Forward-Backward SDE (FBSDE) Equivalence

The primary challenge of solving high-dimensional Partial Differencial Equations(PDEs) is the Curse of Dimensionality. To circumvent this, the Deep FBSNN method leverages the powerful probabilistic representation theorem that connects the solution of a wide class of high-dimensional PDEs to the solution of a Forward-Backward Stochastic Differential Equation (FBSDE) system. The FBSDE is defined by:

$$\begin{aligned} \text{Forward SDE:} \quad & dX_t = b(t, X_t, a_t)dt + \sigma(t, X_t, a_t)dW_t \\ \text{Backward SDE:} \quad & dY_t = -f(t, X_t, Y_t, Z_t)dt + Z_t dW_t \end{aligned}$$

where X_t is the state process and W_t is a Wiener process. For a quasilinear PDE, the function f is related to the PDE nonlinearity. Crucially, the solution $u(t,x)$ of the PDE is related to the solution of the FBSDE by the relation $Y_t = u(t, X_t)$ and $Z_t = \sigma^T \nabla u(t, X_t)$. The objective of the Deep FBSNN is to find the function $u(t,x)$ by training a network to predict Y_t and its gradient proxy Z_t along the simulated paths.

B. Deep FBSNN Framework (Deep BSDE)

The Deep FBSNN (often referred to as the Deep BSDE method) approximates the solution (Y_t, Z_t) by discretizing the time domain $[0,T]$ into N steps. The core idea is to replace the unknown functions $u(t,x)$ and $\nabla u(t,x)$ with deep neural networks.

The time domain $[0,T]$ is discretized into N steps, and the FBSDE is approximated using the Euler-Maruyama scheme over M independent realizations:

$$X_{n+1}^m \approx X_n^m + \mu(t_n, \dots)\Delta t + \sigma(t_n, \dots)\Delta W_n$$

The network is trained to minimize the following loss function, which incorporates the terminal condition and the SDE consistency error along the paths:

$$Loss = \sum_{m=1}^{M} \left(\sum_{n=0}^{N-1} |Y_{n+1}^m - \hat{Y}_{n+1}^m|^2 + |Y_N^m - g(X_N^m)|^2 \right)$$

where Y_{n+1}^m is the neural network prediction at t_{n+1}, and $\hat{Y}_{n+1}^m$ is the one-step SDE-based prediction derived from the network's output at t_n.

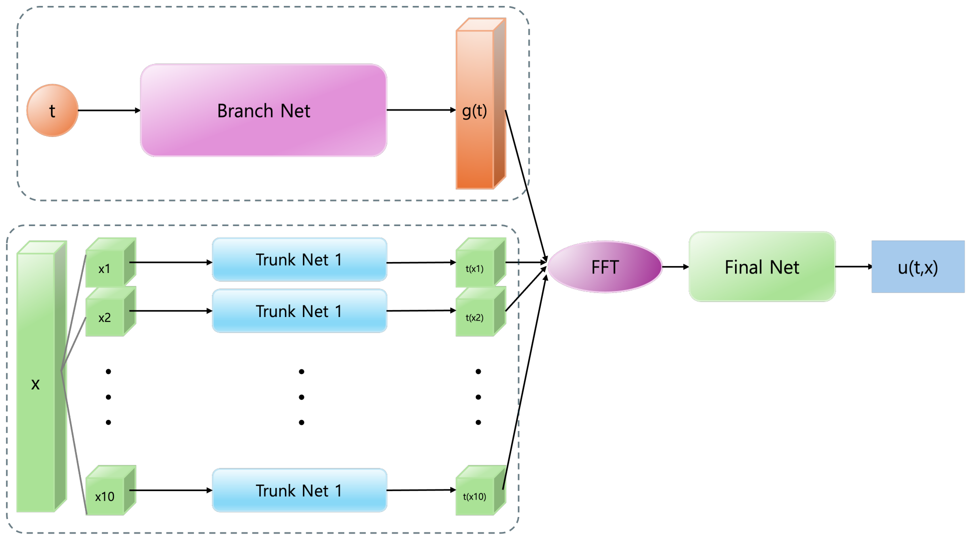

Fig. 1. FourierMultiDeepONet Architecture. This Multi-Head design splits the spatial input x into parallel trunks for computational efficiency and integrates the time features $g(t)$ via concatenation and Fourier Feature Mapping (labeled FFT) before passing to the Final Net.

C. Modification Strategies for Trajectory Accuracy

Despite the success of the standard Deep FBSNN in certain PDEs, this work observed a persistent discrepancy in the internal time trajectory of the HJB solution. To address this, two primary modifications rooted in AI methodologies were implemented.

1) Multi-Head Architecture: FourierMultiDeepONet: To address the perceived lack of temporal dependency and to explore the potential for computational efficiency in the Deep FBSNN framework, I implemented a modified DeepONet-based structure [2] named FourierMultiDeepONet (FMDONet). This architecture was chosen over the standard MLP used in the original FBSNN to enhance the model's ability to represent the complex spatio-temporal function $\mathbf{u}(t, x)$ through modularity and explicit feature mapping.

The FMDONet architecture consists of the following components (in Figure 1):

- Branch Network: This network processes the 1-dimensional time input (t). It is responsible for extracting the time-dependent features $\mathbf{g}(t)$ which serve as coefficients for the basis functions.
- Parallel Trunk Networks: This component handles the high-dimensional spatial coordinates (x). Instead of a single Trunk network, the input x is partitioned and processed by multiple parallel MLP blocks (Multi-Head structure). This design aimed to reduce the number of parameters and allow for computational benefits through complex parallelization (e.g., multi-threading).
- Fourier Feature Mapping: The concatenated outputs of the Branch ($\mathbf{g}(t)$) and Trunk ($\mathbf{t}(x)$) networks are passed through a Fourier Feature Mapping layer. This technique explicitly projects the input into a higher-dimensional space using sinusoidal functions, enabling the subsequent layers to capture fine-grained, high-frequency patterns present in PDE solutions.
- Final Network: A final shallow MLP processes the Fourier-mapped, expanded feature vector to output the scalar prediction, $u(t, x)$.

2) Causal Loss Balancing (Backward Path Formulation): The second strategy involves augmenting the standard FBSNN loss with a Backward Causal Loss, $Loss_{B\text{-causal}}$, designed to stabilize training by prioritizing the minimization of residuals proximal to the terminal condition $t = T$. This approach implicitly forces the network to satisfy the terminal constraint before resolving the dynamics at earlier time steps, drawing inspiration from causality concepts in related deep learning applications [3]. The total loss is defined as the sum of weighted losses over all time intervals, $[t_0, t_N]$, where the weight factor depends on the accumulated future residuals:

$$Loss_{B\text{-causal}} = \sum_{n=0}^{N-1} \left(\exp\left(-\xi \sum_{k=n+1}^{N-1} Loss_k \right) \cdot Loss_n \right)$$

where $Loss_n = E[|Y_{n+1} - \hat{Y}_{n+1}|^2]$ is the Mean Squared Error (MSE) residual loss at segment n. Here, Y_{n+1} is the neural network approximation at time t_{n+1}, $\hat{Y}_{n+1}$ is the SDE-based one-step forward prediction derived from the network's output at t_n, and ξ is the causality parameter controlling the steepness of the weighting. The inner summation, $\sum_{k=n+1}^{N-1} Loss_k$, calculates the cumulative future error used to scale the weight $w_{rev,n} = \exp(-\xi \sum_{k=n+1}^{N-1} Loss_k)$.

IV. Experiments

The effectiveness of the proposed architecture and loss balancing technique was evaluated on the 100-dimensional Hamilton-Jacobi-Bellman(HJB) equation. This problem serves as a critical benchmark where the standard Deep FBSNN method exhibits instability in the internal trajectory due to the equation's strong nonlinearity.

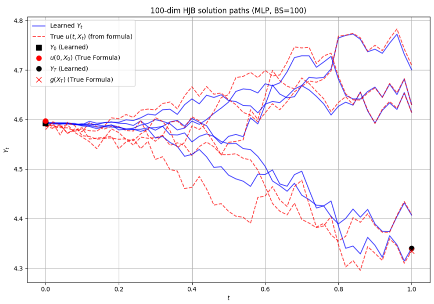

Fig. 2(a). Baseline Performance (Standard MLP). Shows the inherent prediction discrepancy in the intermediate trajectory ($t \in (0, T)$).

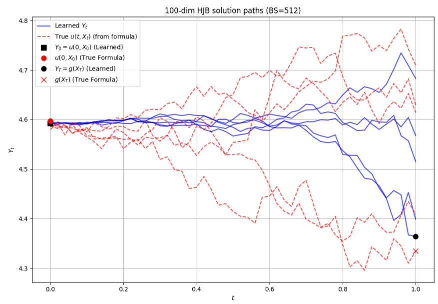

Fig. 2(b). FMDONet Result. The architectural modification failed to significantly reduce the prediction discrepancy compared to the baseline.

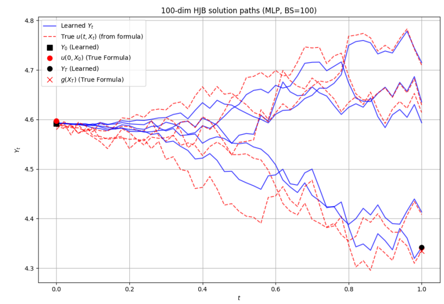

Fig. 2(c). Causality Loss Balancing Result. The loss re-weighting strategy failed to improve the fidelity of the internal solution path.

Fig. 2. Comparison of Learned vs. True HJB Solution Paths (100D). The dashed red lines represent the true solution paths; the solid blue lines represent the learned trajectories. All methods fail to resolve the intermediate trajectory error.

The HJB problem is related to the following Forward-Backward Stochastic Differential Equations (FBSDEs):

$$\begin{aligned} dX_t &= \sigma dW_t, \quad t \in [0, T], \quad X_0 = \xi \\ dY_t &= ||Z_t||^2 dt + \sigma Z_t' dW_t, \quad t \in [0, T) \\ Y_T &= g(X_T) \end{aligned} \tag{1}$$

The corresponding PDE is given by $u_t = -\text{Tr}[D^2 u] + ||Du||^2$, with terminal condition $u(T, x) = g(x)$.

The specific parameters used for this experiment are: Dimension $D = 100$, Time Horizon $T = 1$, Diffusion Coefficient $\sigma = \sqrt{2}$, Initial Condition $\xi = (0, 0, \ldots, 0) \in \mathbb{R}^{100}$, and Terminal Condition $g(x) = \ln(0.5(1 + ||x||^2))$.

A. Solution and Experimental Setup

The ground truth solution $u(t, x)$ for the HJB equation is approximated using 10^5 Monte-Carlo samples due to the presence of the expectation operator in its explicit form. The baseline time discretization was $N = 50$ intervals. Our experiments apply the proposed FourierMultiDeepONet and Causality Loss Balancing methods to this HJB problem.

The following section presents the visual comparison of the internal trajectory prediction performance for the 100D HJB equation, comparing the baseline against our two proposed modification methods.

B. Results and Analysis of Failure

The visual evidence from Figure 1 confirms the consistent failure of both the structural and optimization methods to enhance internal trajectory accuracy. This demands an investigation into the fundamental limitations of the Deep FBSNN framework:

- Structural and Technical Constraint Failure: The Multi-Head structure, while intended for parallelization, acted merely as a sequence of MLPs, failing to introduce a fundamental change in the function representation $\mathbf{u}(t, x)$. Technical constraints further inhibited the realization of expected computational gains.
- Causality Loss Balancing Failure: The loss balancing technique failed due to the Deep FBSNN's core mechanism: the continuous, dynamic sampling of SDE paths at every training iteration. This constant shifting of the input data distribution prevents static loss balancing schemes from achieving stable efficacy.

The analysis suggests the failure is a result of the stochastic, path-dependent nature of the FBSNN being fundamentally incompatible with loss balancing techniques successful in static, grid-based frameworks like PINNs. This underscores the need for future research to focus on revising the core loss structure and the integration of the stochastic processes.

The consistent failure of both proposed methodologies to improve the internal trajectory accuracy of the **100**-dimensional HJB solution necessitates a deeper inquiry into the fundamental constraints of the Deep FBSNN framework. I conclude that the observed performance ceiling is due to an incompatibility between the stochastic training mechanism and the optimization techniques applied.

1) Failure of Structural Enhancement (FourierMultiDeepONet): The architectural failure stemmed from two primary issues:

- Lack of Fundamental Change: The FourierMultiDeepONet structure, despite its use of parallel trunks and Fourier feature mapping, ultimately acted as a sequence of standard Multi-Layer Perceptrons (MLPs). This architecture did not introduce the required fundamental change in the function representation $\mathbf{u}(t, x)$ necessary to encode the complex, path-dependent nature of the HJB solution more effectively than the baseline network.
- Negation of Computational Benefits: The potential gains from the Multi-Head structure—specifically parameter efficiency and training acceleration through complex parallelization—were not realized. This was due to hardware constraints (GPU) and the technical inability to implement optimal multi-threading, which limited the practical advantages of the parallelized design.

2) Failure of Optimization Strategy (Causality Loss Balancing): The failure of the Causality Loss Balancing technique is attributed to a critical conflict between the loss mechanism and the training data generation:

- Dynamic Data Distribution Conflict (Primary Cause): The Deep FBSNN fundamentally relies on sampling new

spatio-temporal coordinates by simulating SDE paths at every training iteration. This means the input data distribution is continuously and dynamically shifting. Consequently, the weighted loss scheme, which requires a stable or predictable optimization landscape, cannot achieve stable efficacy. The effect of the loss balancing is immediately undermined by the shifting data distribution of the next minibatch.
- Dominance of Terminal Constraint: Furthermore, the optimization remains overwhelmingly dominated by the strong, implicit constraint imposed by the terminal condition (Y_T) via the loss function. This terminal constraint effectively absorbs or overwhelms any local adjustments attempted by the Causality Loss Balancing on the intermediate steps ($t \in (0, T)$).

3) Discussion of Results and Limitations: This analysis provides a critical insight that the failure is not merely a matter of hyperparameter tuning, but a result of structural incompatibility. Loss balancing techniques, which are effective in static, grid-based frameworks (like PINNs) where the data distribution is fixed, are fundamentally ineffective within the stochastic and path-dependent FBSNN framework. This issue is further compounded by an inherent structural bias in the standard FBSNN formulation. This underscores the necessity for future research to move away from external weighting schemes and focus instead on revising the core loss structure or finding new methods for the local integration of stochastic processes to achieve robust trajectory prediction. Recent work has specifically addressed the need for improved integration schemes for solving PDEs with Backwards SDEs, demonstrating efforts to both theoretically prove and algorithmically eliminate this fundamental bias [4].

V. Conclusion

This study investigated methods to enhance the internal trajectory accuracy of the solution to the 100-dimensional Hamilton-Jacobi-Bellman (HJB) equation using the Deep Forward-Backward Stochastic Neural Network (Deep FBSNN) framework. A significant prediction discrepancy was consistently observed in the intermediate time domain $t \in (0, T)$ when the standard FBSNN method was applied. We proposed and evaluated two distinct modification strategies: the architectural enhancement via the FourierMultiDeepONet (a Multi-Head structure) and the optimization refinement via the Causality Loss Balancing technique.

Despite these targeted efforts, both methodologies consistently failed to yield significant improvements in trajectory fidelity, confirming the persistence of the intermediate prediction error. The failure was primarily attributed to a fundamental structural incompatibility. The core mechanism of the Deep FBSNN involves continuous dynamic sampling of SDE paths at every training iteration. This shifting data distribution renders static loss weighting schemes, like Causality Loss Balancing, ineffective and unstable, as the optimization landscape is constantly undermined. Furthermore, the FourierMultiDeepONet structure failed to introduce the necessary fundamental change in the complex function representation $\mathbf{u}(t, x)$ required to overcome the strong nonlinearity of the HJB problem, highlighting architectural limitations.

This analysis provides a critical insight: While loss balancing is a successful optimization tool in static, grid-based contexts, it is fundamentally ill-suited for the stochastic and path-dependent nature of the Deep FBSNN framework due to the dynamic data conflict. Furthermore, existing FBSNN structures contain an inherent bias, necessitating a resolution of the core structural issues rather than merely applying optimization techniques.

Therefore, future research should move away from external weighting schemes and prioritize structural changes that directly address the local dynamics of the stochastic process. Potential avenues include revising the core loss structure by developing a loss function derived from a non-local formulation of the SDE that naturally incorporates temporal relationships, rather than relying on external weighting factors. Additionally, exploring the architectural integration of stochastic process constraints (perhaps through more advanced sequential or temporal models) could better encode the path dependency ofY_t and Z_t.

References

[1] M. Raissi, "Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations," 2018.

[2] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, p. 218–229, Mar. 2021.

[3] S. Wang, S. Sankaran, and P. Perdikaris, "Respecting causality is all you need for training physics-informed neural networks," 2022.

[4] S. Park and S. Tu, "Integration matters for learning pdes with backwards sdes," 2025.