A NEURAL NETWORK APPROACH TO HIGH-DIMENSIONAL OPTIMAL SWITCHING PROBLEMS WITH JUMPS IN ENERGY MARKETS

ERHAN BAYRAKTAR, ASAF COHEN, AND APRIL NELLIS

Abstract. We develop a backward-in-time machine learning algorithm that uses a sequence of neural networks to solve optimal switching problems in energy production, where electricity and fossil fuel prices are subject to stochastic jumps. We then apply this algorithm to a variety of energy scheduling problems, including novel high-dimensional energy production problems. Our experimental results demonstrate that the algorithm performs with accuracy and experiences linear to sub-linear slowdowns as dimension increases, demonstrating the value of the algorithm for solving high-dimensional switching problem.

Keywords. Deep neural networks, forward-backward systems of stochastic differential equations, optimal switching, Monte Carlo algorithm, optimal investment in power generation, planning problems

1. Introduction

Energy production and energy markets play a large role in the modern economy and as such, it is beneficial to both producers and consumers for electricity production to be optimized. Energy producers, in particular, desire to operate efficiently despite the inherent volatility of both electricity demand and the availability of various fuels. Determining the correct operating strategy for an energy production facility therefore requires dynamic adjustment as the underlying drivers of price and profit fluctuate stochastically with supply and demand. Recent supply-chain issues in global markets have further underlined the volatility of prices and the need for flexible optimization methods which allow producers to dynamically adapt to changes in the energy markets.

There are multiple perspectives from which to approach problems related to energy production and pricing. In the case where a model includes only a single power generation facility, this facility is considered a price-taker, and its production decisions have little impact on the overall flow of electricity supply and demand. The facility's only goal is to maximize its own profit, as it is not the sole electricity producer in its region. To this end, the facility is able to alter its own production capacity in response to exogenous outside factors. However, we can also consider a situation in which an agent oversees multiple power generation facilities, and has the option to bring them online or remove them. Each of these facilities is fueled by one of a selection of fuel sources, ranging from coal to solar energy. The larger scale of this operation makes this agent a price-setter, and so investment decisions affect both electricity spot prices and their own profits. In this case, penalties could also be incurred for failing to satisfy electricity demand. Our focus will be on the former case, but our algorithm could easily be extended to other situations.

These situations can be modeled as optimal switching problems, and in our paper we present a machine learning algorithm that is able to solve optimal switching problems of higher dimensions than previously studied, allowing us to consider a wider selection of fuel sources than in existing literature. Such energy production switching problems consist of a stochastic state process (such as exogenous electricity demand and fuel prices) which drives an objective function. At discrete "switching times" a production decision is chosen from a discrete set of possible "modes" of production (which can model factors like capacity level or fuel type). The controller switches between modes based on the current value of the state variable, but must pay a penalty for such switches (usually monetary, reflecting resource redirection). The class of optimal switching problems is one that has both been investigated from an analytical perspective [22, 5, 13, 14] and applied to fields from finance [29] to cloud computing [17], but these problems remain difficult to solve numerically in higher dimensions. In the realm of energy markets, mathematicians have used optimal switching to model power plant scheduling [12, 35], electricity spot prices [1], and run-of-river hydroelectric power generation

TO APPEAR IN SIAM JOURNAL ON FINANCIAL MATHEMATICS.

[33]. Energy storage problems [18, 32, 39] are another popular application of optimal switching, but we focus on scheduling and production problems in our current work.

Various approaches have been taken to avoid a grid-based method, as grids are very susceptible to the so-called "curse of dimensionality", including many Monte Carlo-based methods like [40, 1]. However, such probabilistic approaches are also limited in the dimension they can handle, as most rely on regression over a number of basis functions that grows quickly with the dimension of the state space. In recent years, the applications of machine learning to mathematical problems has become more and more common, many inspired by seminal works such as [24], which trains a neural network to minimize the global error associated with the backward stochastic differential equation (BSDE) representation of certain classes of partial differential equations (PDEs). Expanding upon this work, neural networks have been found to accurately estimate the solutions of a variety of partial differential equations of varying complexities when used in different configurations, as in [26, 36, 6, 20]. In addition, the early paper [2] utilized neural networks to solve for an optimal gas consumption strategy under uncertainty. It follows that such neural network-based methods can be extended to solve optimal switching problems. Our algorithm draws upon the neural-network-based deep backward dynamic programming approach introduced in [26] and extends it to situations where the reflection boundary is no longer a known function, like the payoff of an American option. Instead, the reflection boundary becomes dependent on the optimal control decision at the given point in time. We also introduce jumps in the state process, which change the associated formulation from a partial differential equation to a partial integro-differential equation (PIDE). These jumps are incorporated into the model to better simulate the volatility inherent in electricity and fossil fuel markets. The recent work [21] extends [24] to a setting with jumps, and [19] applies neural networks to PIDEs that arise in insurance mathematics.

In this paper, we extend [26] to handle both jumps and switches in a wider range of problems. This algorithm is able to handle high-dimensional problems well because the time needed for artificial neural network computations grows only linearly in the dimension of the state variable and suffers only minimal slowdowns as the dimension increases, as demonstrated in Section 4. Our code can be found at `https://github.com/april-nellis/osj`.

In Section 2, we introduce the general stochastic model of an optimal switching problem. In Section 3, we provide some background on neural networks and detail the proposed machine learning algorithm. In Section 4 we discuss numerical examples of energy scheduling and capacity investment, and demonstrate the high-dimensional abilities of our algorithm[1]. In Section 5, we verify the convergence of the neural networks in our proposed algorithm to the true value functions.

2. Stochastic Model

2.1. **Setup.** The goal of our paper is to numerically solve high-dimensional optimal switching problems related to energy production. Consider a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_t, \mathbb{P})$ satisfying the usual conditions and supporting a d-dimensional Wiener process W and a one-dimensional Poisson random measure $\mathcal{N}(de, ds)$ with intensity measure $\nu(de)ds$, where $\int_{\mathbb{R}^d} \nu(de) = \lambda \geq 0$. Consider further a d-dimensional jump-diffusion process, given by

$$X_t = x_0 + \int_0^t b(X_s)ds + \int_0^t \sigma(X_s)dW_s + \int_0^t \int_{\mathbb{R}^d} \beta(X_{s^-}, e)\mathcal{N}(de, ds),\ t \in [0, T], x_0 \in \mathbb{R}^d. \tag{2.1}$$

Here, $b : \mathbb{R}^d \to \mathbb{R}^d$, $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, and $\beta : \mathbb{R}^d \times E \to \mathbb{R}^d$, where d is a relatively large dimension and $E \subseteq \mathbb{R}^d$.

Assumption 1. *We assume that*

(1) The functions b, σ, and β are Lipschitz, and β is a measurable map such that there exists $K > 0$ for which

$$\sup_{\xi\in E} |\beta(0, \xi)| \leq K \text{ and } \sup_{\xi\in E} |\beta(x, \xi) - \beta(x', \xi)| \leq K|x - x'|,\ \forall x, x' \in \mathbb{R}^d.$$

(2) The function $\beta(x, \xi)$ has Jacobian such that $\nabla\beta(x, \xi) + I_d$ is invertible with a bounded inverse.

Remark 1. *From the appendices of* [8]*, the conditions in assumption 1 imply the existence of a unique adapted solution X_t to eq.* (2.1)*.*

[1]All calculations in this paper were performed on a 10-core CPU, 16-core GPU 2021 Macbook Pro with M1 Pro chip, without using GPU acceleration.

This stochastic process drives an optimal switching problem which we will solve using a series of artificial neural networks. Variations on this problem have been studied in previous theoretical papers such as [23] and [16], and can be summarized as trying to find the optimal choice of control $\boldsymbol{a} = \{(\tau_k, \alpha_k)\}_{k\in\mathbb{N}}$, where $\alpha_k \in \mathbb{I} =: \{1, \dots, I\}$ is the regime/mode which is selected at switching time τ_k, such that α_k is $\mathcal{F}_{\tau_k}$-measurable, for any $k \in \mathbb{N}$. We set $\tau_0 = 0$ to denote that a switch is allowed as soon as the process begins. The initial mode of the system, i, is therefore denoted by $\alpha_{-1} = i$. The control process $(a_s)_{s\in[0,T]}$ associated with $\boldsymbol{a}$ is denoted by:

$$a_s = \sum_k \alpha_k \mathbf{1}_{\{\tau_k \le s < \tau_{k+1}\}}.$$

It represents the current mode of the system and the set of such strategies is given by $\mathcal{A}$. The set of *admissible strategies* is defined as all strategies fitting the above description which contain only a finite, though potentially random, number of switches, and is denoted $\mathcal{A}$. The set of admissible strategies that begin in mode i at initial time t is denoted $\mathcal{A}_{t,i}$. The expected payoff function associated with the control $\boldsymbol{a} \in \mathcal{A}_{t,i}$ is given by

$$J(t,x,i,\boldsymbol{a}) := \mathbb{E}\left[\int_t^T f_{a_s}(s, X_s)ds + g^{a_T}(X_T) - \sum_{k\in\mathbb{N}\setminus\{0\}} C_{\alpha_{k-1},\alpha_k}(X_{\tau_k})\mathbf{1}_{\{t\le\tau_k<T\}}\Big| X_t = x, a_t = i\right], \tag{2.2}$$

where $f_i : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}$ is the running profit in mode i, $g^i : \mathbb{R}^d \to \mathbb{R}$ is the terminal profit if ending in mode i, and $C_{i,j} : \mathbb{R}^d \to \mathbb{R}$ is the cost of switching modes from i to j for a given value of the state variable, where $i, j \in \mathbb{I}$. Here and in the sequel, $\mathbf{1}_B$ is the indicator of the event B, such that $\mathbf{1}_B(\omega) = 1$ if $\omega \in B$ and 0 otherwise. We can define the initial status of the system as $\mathcal{F}_0 := \{X_0 = x_0, \alpha_{-1} = i\}$. All expectations are conditioned on $\mathcal{F}_0$ when not otherwise specified.

Assumption 2. *To discourage an optimal strategy with multiple instantaneous switches, we make the following assumptions on the switching costs. There exists $\epsilon > 0$ such that*

$$\begin{aligned}
&C_{i,j}(x) \ge \epsilon > 0, \ \forall i, j \in \mathbb{I}, \ \forall x \in \mathbb{R}^d,\\
&C_{ii}(x) \equiv 0, \ \forall i \in \mathbb{I},\\
&C_{i,j}(x) + C_{j,k}(x) \ge C_{i,k}(x), \ \forall i, j, k \in \mathbb{I}, \ \forall x \in \mathbb{R}^d.
\end{aligned}$$

These assumptions are standard in optimal switching problems, encoding "direct" switches between states, and are enforced throughout the paper. We also make the Lipschitz assumption that there exists a constant $[C]_l$ such that

$$|C_{i,j}(x_1) - C_{i,j}(x_2)| \le [C]_l||x_1 - x_2||,$$

for all $x_1, x_2 \in \mathbb{R}^d$ and all $i, j \in \mathbb{I}$.

We also make certain assumptions on the running profit and terminal profit functions throughout the paper.

Assumption 3. .

(1) There exists a constant $[f]_l$ such that for every $t_1, t_2 \in [0, T]$ and $x_1, x_2 \in \mathbb{R}^d$,

$$|f_i(t_1, x_1) - f_i(t_2, x_2)| \le [f]_l(|t_1 - t_2|^{1/2} + ||x_1 - x_2||), \ \forall i \in \mathbb{I}.$$

(2) We assume $\max_{i\in\mathbb{I}} \sup_{0\le t\le T} |f_i(t, 0)| < \infty$ and f_i is square-integrable on $[0, T]$ for all i in $\mathbb{I}$.
(3) The functions $\{g^i\}_{i\in\mathbb{I}}$ are Lipschitz continuous and satisfy linear growth conditions.

The value function is given by

$$V(t, x, i) := \sup_{\boldsymbol{a}\in\mathcal{A}_{t,i}} J(t, x, i, \boldsymbol{a}).$$

It is a standard result in control theory (see [15, 7]) that the solution to this optimization problem can be represented as a real-valued stochastic process $Y_t^i = V(t, X_t, i)$ that solves the stochastic differential equation

$$
\begin{aligned}
&Y_t^i = g^i(X_T) + \int_t^T f_i(s, X_s)ds - \int_t^T (Z_s^i)^T dW_s - \int_t^T \int_{\mathbb{R}^d} \Delta Y_s^i(e)\tilde{\mathcal{N}}(de, ds) + R_T^i - R_t^i, \\
&Y_t^i \geq \max_{j\neq i}\{-C_{i,j}(X_t) + Y_t^j\},\ t \in [0,T], \\
&\int_0^T (Y_t^i - \max_{j\neq i}(-C_{i,j}(X_t) + Y_t^j))dR_t^i = 0.
\end{aligned} \tag{2.3}
$$

where $\tilde{\mathcal{N}}(de, ds) := \mathcal{N}(de, ds) - \nu(de)ds$ and the reflection boundary R_t^i is a nondecreasing process with $R_0^i = 0$. Further, the auxiliary processes Z_t^i and $\Delta Y_t^i(e)$ can be defined as

$$
\begin{aligned}
Z_t^i &:= \sigma^T(X_t)V_x(t, X_t, i) \in \mathbb{R}^d, \\
\Delta Y_t^i(e) &:= V\big(t, X_{t^-} + \beta(X_{t^-}, e), i\big) - V(t, X_{t^-}, i) \in \mathbb{R}.
\end{aligned}
$$

The stochastic differential equations for X_t and Y_t comprise a system of forward-backward stochastic differential equations (FBSDEs). In addition, the continuation values associated with beginning in mode i at time t_1 and remaining in that mode over the interval $[t_1, t_2]$ for $t_1, t_2 \in [0, T]$ can be defined via eq. (2.3) as

$$
\tilde{Y}_{t_1}^i := Y_{t_2}^i + \int_{t_1}^{t_2} f_i(s, X_s)ds - \int_{t_1}^{t_2} (Z_s^i)^T dW_s - \int_{t_1}^{t_2} \int_{\mathbb{R}^d} \Delta Y_s^i(e)\tilde{\mathcal{N}}(de, ds).
$$

Approximation of certain continuation values will play a key role in approximating the value function of interest. Our goal in the rest of this paper is to present an efficient algorithm for calculating Y^i where X is a high-dimensional state process with finite-variational jumps. In section 3, we first provide some background on neural networks, then we present the details of the Optimal Switching with Jumps (OSJ) algorithm.

3. Optimal Switching with Jumps (OSJ) Algorithm

3.1. **Neural Network Structure.** We utilize feedforward neural networks, which are in essence a series of weighted sums of inputs composed with simple functions in such a way that unknown functions can be approximated. Training data enters the network in the first layer, and at each layer a weighted sum of the inputs is computed using the choice of parameters assigned to the nodes in that layer to create an affine function. The output of each layer is processed by an activation function before becoming the input of the next layer, and the final layer produces the desired output of the network.

For a network of depth δ with δ_ℓ nodes in layer ℓ, there are $\sum_{\ell=0}^{\delta-1} \delta_\ell(\delta_{\ell+1} + 1) = \bar{\delta}$ parameters, represented as a whole as θ. This θ is chosen from all possible parameters in the parameter space Θ_δ, a compact subset of $\mathbb{R}^{\bar{\delta}}$ defined as

$$
\Theta_\delta := \{\theta \in \mathbb{R}^{\bar{\delta}},\ ||\theta||_\infty \leq \gamma_\delta\},
$$

where γ_δ is positive and chosen to be very large. We can then define the set of neural networks that we are working with as the union over $\delta \in \mathbb{N}$ of all the neural networks of depth δ with $\bar{\delta}$ total parameters. This formulation accomplishes two things. First, the universal approximation theorem of [25] asserts that this set of neural networks is dense in the set of continuous and measurable functions which map from $\mathbb{R}^d \to \mathbb{R}^s$, for any dimension s, and so are universally good approximators. Second, the parameter space associated with this union, $\Theta = \cup_{\delta\in\mathbb{N}}\Theta_\delta$, represents the set of all possible weights that can be assigned to the nodes in the neural network and is compact. Therefore, when trying to minimize the loss function associated with our problem (which will be described in the next subsection), a minimizing θ^* exists.

The network therefore "learns" the function of interest by adjusting θ via multiple iterations of an optimization algorithm. In our work, we use the Adam optimizer [27] applied to a four-layer neural network with $d + 10$ nodes in each layer and *tanh* as the chosen activation function. We fix the input dimension as d, and set the output dimension as $d_1 = 1 + d + 1$ because $Y_t^i \in \mathbb{R}, Z_t^i \in \mathbb{R}^d$, and $\Delta Y_t^i \in \mathbb{R}$.

3.2. **Algorithm.** To perform the numerical calculations, we discretize the continuous time interval $[0, T]$ using a regular grid $\pi = \{t_n\}_{n=0}^M = \{nT/M\}_{n=0}^M$, where $T/M = \Delta t$. We denote the paths of the discrete approximation as X^π and generate a large number of paths of X^π starting from a desired initial condition x_0. We later use these paths as training data for the neural networks. At this point, we do not impose a specific approximation scheme, but require that the in-time convergence be of at least strong order 0.5 for convergence of the neural network value function in theorem 1 and of strong order 1.0 to achieve an auxiliary

result regarding the performance of the neural network-generated strategies given in theorem 2. We describe the approximation schemes used in our specific examples within the expository portions of section 4.1 and section 4.2. In-depth discussion of weak- and strong-order approximations of jump-diffusion processes can be found in [28].

We also discretize the switching times, introducing a grid $\mathfrak{R}$ where the grid spacing is of size $T/\sqrt{M}$, meaning that $|\mathfrak{R}| \sim O(M^{-1/2})$. The process is able to switch modes at time t_n where $t_n \in \mathfrak{R}$, while evolving as an uncontrolled jump-diffusion process when $t_n \notin \mathfrak{R}$.

The continuation values between time steps of the grid π will be learned by the neural network on a mode-by-mode basis. We denote the neural network that learns the continuation value at t_n in mode i as $\mathcal{Y}_n^i(X_n^\pi, \theta_1^i)$, the neural network that learns the derivative of the value function at the same stage as $\mathcal{Z}_n^i(X_n^\pi, \theta_2^i)$, and the neural network that learns the jump sizes for Y as $\Delta\mathcal{Y}_n^i(X_n^\pi, \theta_3^i)$. In practice, these neural networks are treated as one larger network with combined parameter vector $\theta^i = (\theta_1^i, \theta_2^i, \theta_3^i) \in \Theta = \Theta_\delta$ for some δ corresponding to the chosen architecture of the neural network. The functions generated by the optimal choice of θ^i are defined as

$$\begin{cases} \tilde{\mathcal{Y}}_n^i(X_n^\pi) := \mathcal{Y}_n^i(X_n^\pi, \theta_{n,1}^{*,i}), \\ \hat{\mathcal{Z}}_n^i(X_n^\pi) := \mathcal{Z}_n^i(X_n^\pi, \theta_{n,2}^{*,i}), \\ \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi) := \Delta\mathcal{Y}_n^i(X_n^\pi, \theta_{n,3}^{*,i}). \end{cases} \tag{3.1}$$

The continuation values $\tilde{\mathcal{Y}}_n^i(\cdot)$ are then used to calculate the value functions

$$\hat{\mathcal{Y}}_n^i(X_n^\pi) := \mathbf{1}_{t_n \in \mathfrak{R}} \max\left\{\tilde{\mathcal{Y}}_n^i(X_n^\pi), \max_{j \neq i}(\tilde{\mathcal{Y}}_n^j(X_n^\pi) - C_{i,j}(X_n^\pi))\right\} + \mathbf{1}_{t_n \notin \mathfrak{R}} \tilde{\mathcal{Y}}_n^i(X_n^\pi) \tag{3.2}$$

The algorithm in its entirety is described in algorithm 1.

Algorithm 1 OSJ Algorithm

1: Generate paths of the stochastic process $\{X_n^\pi\}_{n=0}^M$ as well as $\Delta W_n := W_{t_{n+1}} - W_{t_n}$ and $\Delta\tilde{N}_n := \int_{t_n}^{t_{n+1}} \int_{\mathbb{R}^d} \tilde{\mathcal{N}}(de, ds)$ for each sample path. Store as training data.
2: Train $\hat{\mathcal{Y}}_M^i \equiv g^i(x),\ \forall i \in \mathbb{I}$.
3: **for** $n = M-1, \ldots, 2, 1, 0$ **do**
4: **for** all $i \in \mathbb{I}$ **do**
5: Train a neural network to find $\theta_n^{*,i} = (\theta_{n,1}^{*,i}, \theta_{n,2}^{*,i}, \theta_{n,3}^{*,i}) \in \Theta$ which minimizes

$$\begin{aligned} L_n^i(\theta) = \mathbb{E}\Big| & \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi) - \mathcal{Y}_n^i(X_n^\pi, \theta_1) \\ & + f_i(t_n, X_n^\pi)\Delta t - \mathcal{Z}_n^i(X_n^\pi, \theta_2)\Delta W_n - \Delta\mathcal{Y}_n^i(X_n^\pi, \theta_3)\Delta\tilde{N}_n\Big|^2. \end{aligned} \tag{3.3}$$

6: Define $\tilde{\mathcal{Y}}_n^i(\cdot)$, $\hat{\mathcal{Z}}_n^i(\cdot)$, and $\widehat{\Delta\mathcal{Y}}_n^i(\cdot)$ as in eq. (3.1).
7: **end for**
8: **for** all $i \in \mathbb{I}$ **do**
9: Calculate $\hat{\mathcal{Y}}_n^i(\cdot)$ as in eq. (3.2).
10: **end for**
11: The value function of interest is $V(0, x_0, i) = \hat{\mathcal{Y}}_0^i(x)$ where $X_0 = x$ and $\alpha_{-1} = i$.
12: **end for**

The switching strategy arising from this algorithm is denoted $\boldsymbol{a^{NN,M}}$ when the number of steps is chosen to be M. This strategy is a function of the value of the state variable and starting mode, such that the optimal mode at time t_n is

$$\alpha_n^{NN} := \arg\max_{j \in \mathbb{I}} \left(\tilde{\mathcal{Y}}_n^j(\cdot) - C_{\alpha_{n-1}^{NN}, j}(\cdot)\right),$$

where the optimal mode at time t_{n-1} is $\alpha_{n-1}^{NN} \in \mathbb{I}$ and $\alpha_{-1}^{NN} = \alpha_{-1} = i$, the starting mode.

Remark 2. *For theorem 2 we require that the discrete approximation of $(X_t)_{t\in[0,T]}$ is of strong order 1.0 (instead of strong order 0.5 which is sufficient for theorem 1). This excludes the standard Euler–Maryama discretization strategy, but there are a variety of other options. One good choice is a jump-adapted strong*

order 1.0 discretization scheme, first described in [34]*, and other efficient schemes are described in* [30] *and* [31]*. A survey of strong approximations of jump-diffusions is given in* [10] *and expanded upon in chapters 6 and 8 of* [11]*. Since the path generation is done at the start of the algorithm and the paths can be stored for re-use, the downside of increased complexity is outweighed by the benefit of improved convergence rates. In addition, if paths can be simulated exactly (as is the case of the example in section 4.1), the discretization error in X can be avoided altogether, though there remains discretization error from the discrete switching grid.*

3.3. **Convergence.** The natural questions are (1) whether this algorithm will train a neural network that converges to the correct value function of interest, and (2) how the switching strategy generated by this algorithm performs in comparison to the true optimal switching strategy. We first focus on the former question.

The mode-wise maximum of the errors between the FBSDE system $(Y_t^i, Z_t^i, \Delta Y_t^i)$ and the functions learned by the neural network $(\hat{\mathcal{Y}}_n^i, \hat{\mathcal{Z}}_n^i, \widehat{\Delta\mathcal{Y}}_n^i)$, can be represented by

$$\begin{aligned}\mathcal{E}[(\hat{\mathcal{Y}}, \hat{\mathcal{Z}}, \widehat{\Delta\mathcal{Y}})), (Y, Z, \Delta Y)] := &\max_{n=0,1,\dots,M} \mathbb{E}\Big[\max_{i\in\mathbb{I}} |Y_{t_n}^i - \hat{\mathcal{Y}}_n^i(X_n^\pi)|^2\Big] \\ &+ \sum_{n=0}^{M-1} \int_{t_n}^{t_{n+1}} \max_{i\in\mathbb{I}} \mathbb{E}||Z_t^i - \hat{\mathcal{Z}}_n^i(X_n^\pi)||^2 dt \\ &+ \sum_{n=0}^{M-1} \int_{t_n}^{t_{n+1}} \max_{i\in\mathbb{I}} \mathbb{E}\left|\int_{\mathbb{R}^d} \Delta Y_t^i(e)\nu(de) - \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi)\right|^2 dt.\end{aligned} \tag{3.4}$$

We also define auxiliary functions which are integral to the upcoming error definition and to the proofs of convergence themselves. These functions are formulated in a similar manner to that of the discretization of the FBSDEs presented in [8] and are defined as

$$\hat{y}_n^i(X_n^\pi) := \mathbb{E}\left[\hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|\mathcal{F}_n\right] + f_i(t_n, X_n^\pi)\Delta t, \tag{3.5}$$

$$\hat{z}_n^i(X_n^\pi) := \frac{1}{\Delta t}\mathbb{E}\left[\hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)\Delta W_n|\mathcal{F}_n\right], \tag{3.6}$$

$$\hat{u}_n^i(X_n^\pi) := \frac{1}{\lambda\Delta t}\mathbb{E}\left[\hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)\Delta\tilde{N}_n|\mathcal{F}_n\right], \tag{3.7}$$

where $\mathcal{F}_n := \mathcal{F}_{t_n}$ for convenience. According to the martingale representation theorem of [37], there exist processes $(z_s^i)_{t_n<s\le t_{n+1}}$ and $(u_s^i)_{t_n<s\le t_{n+1}}$ such that

$$\hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi) = \hat{y}_n^i(X_n^\pi) - f_i(t_n, X_n^\pi)\Delta t + \int_{t_n}^{t_{n+1}} z_s^i dW_s + \int_{t_n}^{t_{n+1}}\int_{\mathbb{R}^d} u_s^i(e)\tilde{\mathcal{N}}(de, ds). \tag{3.8}$$

Using the Markov property of the processes and Itô isometry, we can also see that the following relationships hold:

$$\hat{z}_n^i(X_n^\pi) = \frac{1}{\Delta t}\mathbb{E}\left[\int_{t_n}^{t_{n+1}} z_s^i ds|\mathcal{F}_n\right], \tag{3.9}$$

$$\hat{u}_n^i(X_n^\pi) = \frac{1}{\lambda\Delta t}\mathbb{E}\left[\int_{t_n}^{t_{n+1}}\int_{\mathbb{R}^d} u_s^i(e)\nu(de)\ ds|\mathcal{F}_n\right]. \tag{3.10}$$

Using these quantities, we define the set of neural network approximation errors as

$$\begin{aligned}\varepsilon_n^y &:= \sum_{i\in\mathbb{I}}\left(\inf_{\theta_1\in\Theta}\mathbb{E}|\hat{y}_n^i(X_n^\pi) - \mathcal{Y}_n^i(X_n^\pi, \theta_1)|^2\right), \\ \varepsilon_n^z &:= \sum_{i\in\mathbb{I}}\left(\inf_{\theta_2\in\Theta}\mathbb{E}||\hat{z}_n^i(X_n^\pi) - \mathcal{Z}_n^i(X_n^\pi, \theta_2)||^2\right), \\ \varepsilon_n^u &:= \sum_{i\in\mathbb{I}}\left(\inf_{\theta_3\in\Theta}\mathbb{E}|\hat{u}_n^i(X_n^\pi) - \Delta\mathcal{Y}_n^i(X_n^\pi, \theta_3)|^2\right).\end{aligned} \tag{3.11}$$

These errors converge to 0 as the number of parameters in the neural network increases to infinity. More precisely, we can say that as $\delta\to\infty$ and correspondingly $|\Theta_\delta|\to\infty$, then $\varepsilon_n^y, \varepsilon_n^z, \varepsilon_n^y$ converge to 0. This

follows from the Universal Approximation Theorem of [25]. Therefore, we can use eq. (3.11) to express our convergence results as a function of the neural network approximation errors and the time discretization M.

Theorem 1 (Convergence of OSJ Value Function)**.** *The error between the machine learning-generated functions $(\hat{\mathcal{Y}}^i, \hat{\mathcal{Z}}^i, \widehat{\Delta\mathcal{Y}}^i)$, and the functions associated with the true optimal switching problem $(Y^i, Z^i, \Delta Y^i)$ (as defined in eq. (3.4)), is bounded by*

$$\mathcal{E}[(\hat{\mathcal{Y}}, \hat{\mathcal{Z}}, \widehat{\Delta\mathcal{Y}})), (Y, Z, \Delta Y)] \leq C \sum_{n=0}^{M-1} \left(M\varepsilon_n^y + \varepsilon_n^z + \varepsilon_n^u\right) + C^\varepsilon \left(M^{-1/2+\varepsilon}\right), \tag{3.12}$$

where $\varepsilon_n^y, \varepsilon_n^z, \varepsilon_n^u$ are defined in eq. (3.11). In addition, $\varepsilon > 0$ is chosen to be arbitrarily small and C^ε has an inverse relationship with ε. The formulation of this error term is further discussed in section 5.1.

Remark 3. *In general, we write $f(x) = O(g(x))$ if there exists some $C > 0$, dependent only on the specific formulation of the problem and not on the size of the discretized grid, such that $f(x) \leq Cg(x)$ as $x \to 0$.*

Remark 4. *Note that our convergence scheme recovers a similar error to Theorem 4.1 of [26] which we extended, with the caveat that the discrete approximation of the multidimensional reflected BSDE eq. (2.3) has size $C^\varepsilon M^{-1/2-\varepsilon}$ instead of $O(M^{-1})$ for their one-dimensional BSDE.*

Remark 5. *The quantity $M\varepsilon_n^y + \varepsilon_n^z + \varepsilon_n^u$ will reappear multiple times as we proceed through the paper. In the future, for convenience we will define*

$$\boldsymbol{\varepsilon_n^M} := M\varepsilon_n^y + \varepsilon_n^z + \varepsilon_n^u,$$

and at times we will use this simpler quantity in place of the sum of the neural network errors.

We also wish to evaluate the performance of $\boldsymbol{a^{NN,M}}$ (the switching strategy generated by the machine-learning algorithm) as M increases to infinity by looking at $J(0, x_0, i, \boldsymbol{a^{NN,M}})$ as defined in eq. (2.2). To do this, we introduce some additional conditions which are not necessary for the results given in theorem 1.

Theorem 2 (Convergence of OSJ Switching Strategy)**.** *Assume that the discrete approximation of $(X_t)_{t\in[0,T]}$ is of strong order 1.0 at minimum. Then, the mode-wise maximum of the errors between the neural network switching strategy's expected payoff $J(0, x_0, i, \boldsymbol{a^{NN,M}})$ and the true value function $V(0, x_0, i)$ is bounded by*

$$\begin{aligned}\max_{i\in\mathbb{I}} |V(0, x_0, i) - J(0, x_0, i, \boldsymbol{a^{NN,M}})| \leq & C^\varepsilon(M^{-1/2+\varepsilon}) + C_2 \sum_{n=0}^{M-1} \left[\frac{\boldsymbol{\varepsilon_n^M}}{M} + \boldsymbol{\varepsilon_n^M}\right] \\ & + O\left(\frac{1}{M^2}\right) \sqrt{O(1) + C_5 \sum_{n=0}^{M-1} \left[\left(\boldsymbol{\varepsilon_n^M}/M\right)^{1/2} + \boldsymbol{\varepsilon_n^M}\right]}.\end{aligned} \tag{3.13}$$

Remark 6. *From [25] it is known that $\varepsilon_n^y, \varepsilon_n^z, \varepsilon_n^u \to 0$ as $|\Theta_\delta| \to \infty$. Therefore, for every M there exists a neural network configuration such that all of these errors can be made arbitrarily small. This implies that for each choice of M, $\boldsymbol{\varepsilon_n^M}$ can also be made arbitrarily small. Specifically, for any $\zeta > 0$, there exists a neural network configuration such that $\boldsymbol{\varepsilon_n^M}$ is on the order of $O(M^{-1-\zeta})$ for all $n \in \{0, \ldots, M-1\}$. This will ensure the convergence of the results shown in this section.*

Remark 7. *We wish to make a small comment on the use of the expectation in our convergence results despite calculating an empirical mean in our algorithm. We choose the number of simulated paths to be on the order of M^2 to ensure that the error between the empirical mean and analytical expectation is reasonably small. Furthermore, if exact simulations can be used, this error does not depend on the time discretization. We refer to Theorem 6.2 of [9] for further reading on this distinction, but omit this error elsewhere in our paper for simplicity.*

The full proofs of theorem 1 and theorem 2 are provided in Section 5.

4. Numerical Examples

4.1. **Optimal Asset Scheduling (Carmona and Ludkovski) [12].** We first numerically evaluate the OSJ algorithm by implementing an optimal switching problem first discussed in [12]. The problem models a power plant that converts natural gas to electricity, where investors are able to purchase three-month lease

contracts. We wish to know how to price these contracts, which is equivalent to calculating the expected maximum profit of the plant over the time horizon under no-arbitrage assumptions.

The price process informing the power plant scheduling decisions has the form of an exponential Ornstein-Uhlenbeck process with jumps given by

$$dX_t = X_t \left[\kappa(\mu - \log(X_t))dt + \Sigma dW_t + \int_{\mathbb{R}^d} (\exp(e) - 1)\mathcal{N}(de, dt)\mathbf{e_1} \right]$$

where $\kappa, \mu, dW_t \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$. The jumps are driven by $\mathcal{N}(de, dt)$, a Poisson random measure with intensity measure $\nu(de) = \lambda\mu(e)de$, where λ is a constant and $\mu(e)$ is the probability density function of e. The standard basis vector $\mathbf{e_1} = (1, 0, \ldots, 0)^T$ is used to denote that only the first dimension (electricity price) experiences a jump.

We use a jump-adapted exact scheme to model the process, where jump times are simulated and added to π to form a new grid, π', with time steps $\{\tau_m\}_{m=0}^{M'}$ where $M' \geq M$. If we define $\tilde{X}_t = \log(X_t)$, then in between grid points in the jump-adapted scheme, the process evolves as a pure diffusion according to

$$d\tilde{X}_t = \left[\kappa(\mu - \tilde{X}_t) - \frac{1}{2}\mathrm{Tr}(\Sigma\Sigma^T) \right] dt + \Sigma dW_t, t \in (\tau_m, \tau_{m+1}),$$

and the exact solution of this Ornstein-Uhlenbeck process is well-known. The sizes of the randomly-occurring jumps is simulated according to the distribution of the exponential random variable e.

4.1.1. *Two-dimensional Example.* The price processes for the electricity sold by the plant and the natural gas bought by the plant are given by the following stochastic processes:

$$\begin{aligned} dP_t &= P_t \left\{ 5(\log(50) - \log P_t)dt + 0.5dW_t^1 + \int_{\mathbb{R}} (\exp(e) - 1)\mathcal{N}(de, dt) \right\}, \\ dG_t &= G_t\{2(\log(6) - \log G_t)dt + 0.4(0.8dW_t^1 + 0.6dW_t^2)\}, \end{aligned}$$

where e follow an exponential distribution where the mean jump size is $1/10 = 10\%$ (and μ is set acccordingly), and the Poisson random measure has $\lambda = 8$, meaning that the average number of jumps is 8 per year. These values are chosen to be consistent with those in [12]. As previously discussed, the power plant can be leased for three-month intervals. Decisions regarding production capacity can be made twice daily, so there are 180 operational decisions to be made over the lifetime of the contract. As in [12], the switching costs are purely dependent on the price of natural gas, as we assume that a certain amount of fuel is burned when altering the operating state of the plant. In this situation, we define

$$C_{i,j} = \begin{cases} 0, & i = j, \forall i, j \in \mathcal{I}, \\ 0.01G_t + 0.001, & i \neq j, \forall i, j \in \mathcal{I}, \end{cases} \tag{4.1}$$

which satisfies the assumptions in assumption 2. For our example, the power plant is assumed to begin the contract period in a dormant state, after which the controller may choose whether to run the plant at full capacity (mode 3), half capacity (mode 2), or turn it off temporarily (mode 1). Each mode has an associated running profit, described by

$$\begin{aligned} f_1(P_t, G_t) &= -1, \\ f_2(P_t, G_t) &= 0.438(P_t - 7.5G_t) - 1.1, \\ f_3(P_t, G_t) &= 0.876(P_t - 10G_t) - 1.2, \end{aligned} \tag{4.2}$$

where the total capacity of the plant is 876 MW and the heat rates are 7.5 MMBtu/MWh when running at half capacity and 10 MMBtu/MWh when running at full capacity.

While this example is only two-dimensional, we implement it to verify the accuracy of our OSJ algorithm on an example that is numerically tractable using previous probabilistic methods. We have made some changes to both the operational setup and the price process evolution in comparison to [12], so we cannot directly compare with the results obtained in that paper. However, we are able to compare the results of our implementation with the results of our implementation of their Longstaff-Schwartz probabilistic algorithm, and find that the discrepancy is very low between the methods. The results of our numerical experiments are displayed in table 1. The visualizations of the switching strategies can be found in fig. 1, and examination shows that the strategies make sense in the context of the problem. For higher natural gas costs and lower

$x_0 = [50, 6]$	OSJ	LS	Difference
$V(0, x_0, 1)$	0.633266	0.63232	0.01%
$V(0, x_0, 2)$	0.69657	0.69683	0.0004%
$V(0, x_0, 3)$	0.621001	0.62914	1.29%

Table 1. This table compares the results of the proposed OSJ algorithm with our implementation of the Longstaff-Schwarz (LS) algorithm presented in [12] to verify the accuracy of the results. 100,000 paths of $\{X_n^\pi\}_{n=0}^{180}$ were generated. The neural networks were trained with a learning rate of 0.001 over 20 epochs, and had a runtime of of 13.73 minutes.

$x_0 = [50, 6]$	OSJ	LS	Difference
$V(0, x_0, 1)$	1.14020	1.14641	0.5%
$V(0, x_0, 2)$	1.19858	1.20930	0.8%
$V(0, x_0, 3)$	1.13858	1.13399	0.4%

Table 2. This extends the results of table 1 to a scenario where $\lambda = 16$ instead of $\lambda = 8$. Similarly, 100,000 paths of $\{X_n^\pi\}_{n=0}^{180}$ were generated and the networks were trained with a learning rate of 0.001 over 20 epochs, and had a runtime of 13.96 minutes.

electricity prices, the plant will have less incentive to produce large amounts of electricity, since the profits of selling electricity will not fully cover the input costs.

Readers may wonder about the performance for higher value of λ. If we double λ from 8 to 16, we obtain similar error estimates, as detailed in table 2, and runtimes are unaffected.

4.1.2. *Higher-dimensional Performance.* We can also consider a high-dimensional example where optimization is done over the price of electricity $X_t^0 = P_t$ (a jump-diffusion process as in the previous example), as well as F possible fuels $(X_t^1, \dots, X_t^F)$, leading to a problem with $d = 1 + F$. The fuel prices $X_t^\phi, \phi = 1, \dots, F$ follow an Ornstein–Uhlenbeck-type stochastic evolution given by

$$dX_t^\phi = X_t^\phi\{2(\log(6) - \log X_t^\phi)dt + 0.4(0.8dW_t^1 + 0.6dW_t^\phi)\}.$$

The modes remain the same (mode 1: shut down, mode 2: half capacity, mode 3: full capacity) and the switching costs use the average over all the input fuels. The profit functions are given by $f_i(P_t, \bar{X}_t)$, defined as in eq. (4.2), where $\bar{X}_t$ is the geometric mean of the fuel prices $X_t^1, \dots, X_t^F$ and has the same distribution as G_t. Therefore, analytically this problem can be simplified to the two-dimensional example, and the output can be compared to that of section 4.1.1. However, the neural network does not "know" that the problems are equivalent, and simply trains the optimal weights for a higher-dimensional input vector and a larger neural network (recall there are $d + 10$ nodes in each layer of the network). So, we can use this example to examine the numerical accuracy and computational performance in higher dimensions. Our results are displayed in table 3, where we demonstrate both consistent accuracy and a linear (rather than exponential) increase in running time in problems of up to 70 dimensions. The increase in running time can also be observed visually in fig. 2.

4.2. **Optimal Capacity Decisions (Aid, Campi, Langrene, Pham** [1]**).** We now consider a high-dimensional example adapted from the paper [1]. In their paper, they investigated the question of when and how many power plants of various types to build over a forty-year time horizon. Specifically, they considered an agent who is able to build power plants which rely on a cheaper fuel or a more expensive one. The optimal construction schedule was able to decrease electricity prices while also maximizing profits. We seek to answer a related question on a shorter time horizon, where the investor owns multiple preexisting power generation facilities that produce electricity using different fuels and sources. The operator must determine the optimal configuration of these facilities to bring online at any given time in order to optimize the profits made by the operator of the facilities. However, electricity prices are also stochastic, and changes in plant

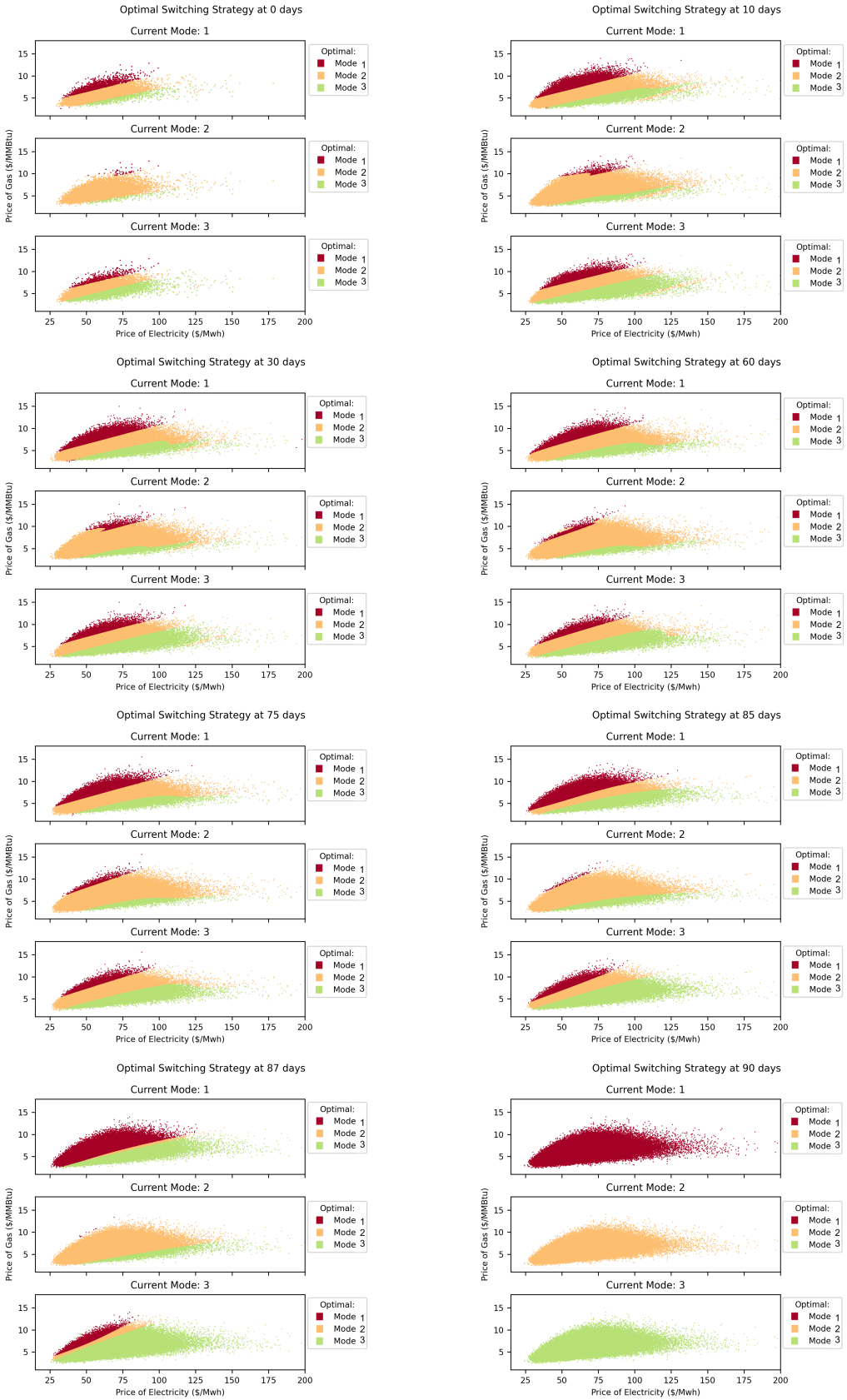

Figure 1. Switching strategies starting from mode $i = 1, 2, 3$ for a selection of time steps, where $N = 180$, corresponding to two electricity production decisions being made per day. The first plot of each subfigure shows the optimal mode to switch to when starting in mode 1 at the given time. The second plot corresponds to starting in mode 2 at the given time, and the third plot corresponds to starting in mode 3 at the given time. Each group of three plots corresponds to a different point in the 90-day scheduling problem.

Dimension (d)	Running time (min)	Average Difference
2	13.73	0.433%
10	19.52	1.004%
20	36.26	1.966%
30	40.05	1.120%
40	51.02	1.231%
50	51.89	1.275%
60	65.82	1.188%
70	71.95	1.617%

Table 3. This table contains the running times and errors for a series of high-dimensional examples which are analytically equivalent to the two-dimensional example solvable by the Longstaff-Schwartz algorithm. By average difference, we refer to the average over the three modes of the differences between the value functions produced by the OSJ and LS algorithms.

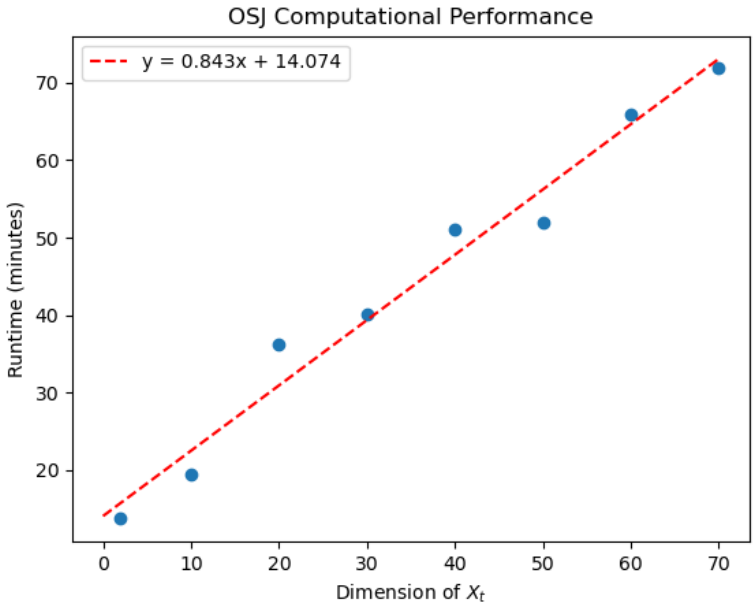

Figure 2. A linear regression of the performance of the algorithm as the dimension of the problem increases.

operations incur switching costs. These changes are made because we would like to investigate strategies on shorter time horizons with more frequent operating decisions, where bringing plants on- and offline in response to demand is not a trivial process. Additionally, the neural network structure of our algorithm will allow us to consider higher-dimensional problems, and so we take advantage of that to model stochastic electricity prices as electricity producers do not always have full control over electricity prices.

One feature of this example is a stochastic "realized availability rate" for each facility, which can be explained as an indicator of fluctuations in realized electricity production due to exogenous factors affecting plant efficiency, despite each facility having a constant "official" output capacity. Additionally, a carbon dioxide emission penalty is included in the price of each input fuel, where the price of carbon dioxide is also stochastic. If we consider F fuel sources (for example, natural gas and oil), the state variables are their

prices $(S_t^\phi), \phi = 1, \dots, F$, their stochastic availability $(A_t^\phi), \phi = 1, \dots, F$, the price of carbon dioxide (S_t^0), the demand for electricity (D_t), and the price of electricity (P_t). Therefore, the total dimension of the state variable is $d = 2F + 3$ where F is the number of different fuels (or more generally, energy sources) being considered. The "modes" we consider are the output levels for each of these fuel sources at the given decision period. We will now proceed to describe the evolution of each of these components of the state variable.

Electricity demand D_t is represented by an Ornstein–Uhlenback process Z_T^0 that is shifted using a cosine function $\mathcal{H}$ to incorporate seasonal demand fluctuations, so that

$$\begin{aligned} dZ_t^0 &= \alpha^0 Z_t^0 dt + \beta^0 dW_t^0, \\ D_t &= Z_t^0 + \mathcal{H}(t). \end{aligned}$$

Realized availability $(A_t^\phi,\ \phi = 1, \dots, F)$ also fluctuates according to an Ornstein–Uhlenback process, and is transformed by a quantile function $\mathcal{T} : \mathbb{R} \to [0, 1]$ to a percentage of the "official" capacity of each power generation facility, as described by

$$\begin{aligned} A_t^\phi &= \mathcal{T}(Z_t^\phi), \\ dZ_t^\phi &= \alpha^\phi Z_t^\phi dt + \beta^\phi dW_t^\phi. \end{aligned}$$

The remaining inputs to the optimal switching problem are the costs of the fuels and the electricity spot prices. In reality, spot prices are set through electricity markets involving complicated interactions between multiple players, but here we model them as stochastic and independent of the operator's switching decision. Fuel costs and electricity prices are correlated through the cointegration matrix μ and covariance matrix Σ, where μ is a matrix with rank r such that $1 < r < F + 2$. The raw fuel prices $S_t^\phi,\ \phi = 0, 1, \dots, F$ and electricity spot prices P_t are jointly referred to as $S_t = (S_t^0, ..., S_t^F, P_t) \in [0, \infty)^{F+2}$, the dynamics of which are given by

$$dS_t = \mu S_t dt + \operatorname{diag}(S_t) \left(\Sigma dW_t^S + \int_{\mathbb{R}^F} S_t(\exp(e) - 1) \mathcal{N}(de, dt) \right).$$

This price process can also be simulated exactly. The electricity price process (the first dimension of the vector) experience jumps of size $S_t(\exp(e)-1)$ where $\mathcal{N}(de, dt)$ is a Poisson random measure with intensity λ and e follows an exponential distribution. The total cost of each fuel is the raw cost of the fuel times its heat rate, $h_\phi S_t^\phi$, plus a carbon dioxide emission charge $h_\phi^0 S_t^0$, and we denote this price as $\tilde{S}_t^\phi = h_\phi^0 S_t^0 + h_\phi S_t^\phi$. The specific values of these parameters are given in table 5. All these factors combine with the current available capacities for each fuel type, given by $K_t^\phi = A_t^\phi \times M_t^{\phi,i}$, where $M_t^{\phi,i}$ is the operating level for the plant that uses fuel ϕ in mode i. We denote the total production capacity at time t by $\bar{K}_t = \sum_{\phi=1}^F K_t^\phi$.

We consider a time horizon $T = 0.25$ years (3 months), where production decisions can be made once per day ($N = 90$). The electricity spot prices, along with the cost of altering the capacity of each type of power generation facility and the current demand, determine the total profit made by the owner of the facilities. The associated running cost is defined as total revenue minus the costs of operating the plants for each technology at the chosen capacities. This can be represented as

$$f(t, X_t, K_t) = P_t \min\{D_t, \bar{K}_t\} + 0.5 P_t (\bar{K}_t - D_t)^+ - 2 P_t (D_t - \bar{K}_t)^+ - \sum_{\phi=1}^F K_t^\phi \tilde{S}_t^\phi,$$

where $X_t = (D_t, A_t^1, \dots, A_t^F, S_t^0, \tilde{S}_t^1, \dots, \tilde{S}_t^F)$ and $K_t = (K_t^1, \dots, K_t^F)$. The basic intuitions behind this profit function are

- Electrical demand must be met.
- The power plant can sell electricity in excess of demand at a steep discount, and can buy additional electricity to satisfy demand at a steep premium.
- The total operational capacity is constant across all modes, but the cost and flexibility of production differs depending on the configuration of the three plants.
- Once mode i is chosen, the power plant must make $K_t^\phi = A_t^\phi \times M_t^{\phi,i}$ units of electricity using technology ϕ.

To demonstrate the expanded capabilities of our algorithm, we consider three different types of power plants: natural gas ($\phi = 1$), coal ($\phi = 2$), and nuclear ($\phi = 3$) plants. Switching costs are given by

$$C_{i,j}(X_t) = \sum_{\phi=1}^{F} \boldsymbol{c}^{\phi} S_t^{\phi} \mathbf{1}_{\{M_t^{\phi,i} \neq M_t^{\phi,j}\}} + \epsilon,$$

where $\boldsymbol{c}^{\phi}$ imposes a scale factor on the cost of fuel ϕ, and there is an additional small fixed switching cost ϵ to satisfy assumption 2. The natural gas-based electricity production facility can be "scaled up" slightly, at the expense of efficiency (conveyed through heat rate), and the "switching cost" of this change is determined accordingly. The coal-based and nuclear production facilities are supplementary facilities that can be turned on or off. Nuclear energy is an interesting addition to the problem described in [1] as it has lower and more consistent running costs than fossil fuels. In addition, nuclear energy production is not subject to a carbon dioxide emissions charge and the cost of uranium is assumed to be only very weakly correlated with fossil fuel pricing. However, the nuclear plant faces high friction when starting up or shutting down the nuclear reactor, as this is a slow and complicated process if done safely. Therefore, we can model this type of plant and its associated security concerns with a higher "switching" cost, quantifying the time and energy required to turn the plant on and off. Finally, the coal plant also takes time and fuel to turn on and off, which is reflected by "switching" cost associated with bringing the plant online or taking it offline. The possible electricity production configurations (modes) that we consider and their switching costs are listed in table 4. Notice that capacity levels are the same for all modes, but the mix of energy sources varies from mode to mode. This allows us to examine the optimal electricity production configuration based on our parameter choices, listed in table 5.

Fuel	$M^{\phi,1}$	$M^{\phi,2}$	$M^{\phi,3}$	$M^{\phi,4}$	Switching cost
Natural Gas ($\phi = 1$)	50	60	60	70	$0.1S_t^1$
Coal ($\phi = 2$)	10	0	10	0	$0.1S_t^2$
Nuclear ($\phi = 3$)	10	10	0	0	$0.5S_t^3$

Table 4. The modes considered in this example involve natural gas ($\phi = 1$), coal ($\phi = 2$), and nuclear ($\phi = 3$) energy sources. $M^{\phi,i}$ is the baseline production capacity of a plant which uses fuel ϕ, when operating in mode i for $i = 1, \ldots, 4$.

The strategies associated with this model are investigated in Figures 3 and 4, which display the relationships between fuel prices and optimal strategies. In each heatmap, fuel prices for one of the three fuels are held constant at their average level, and prices and strategies for the remaining two fuels are visualized. The heatmaps are presented in groups of four to illustrate how the optimal switching strategy is dependent on the current mode at time t_n, and each subfigure shows the optimal strategies (given by the colors in the heatmap) for different values of the state variables at a given time and current mode. This allows us to isolate how prices of natural gas, coal, and uranium individually affect optimal strategies and also to examine how switching costs contribute to switching decisions.

Overall, our parameters defined coal as the least efficient of the three sources, and is also the most expensive once the carbon dioxide emissions charge is incorporated into the pricing. Nuclear energy is the most cost-effective on average, but natural gas is the most efficient when comparing heat rates. Therefore, modes 1 and 3, which utilize the coal plant, are generally less preferred than the others. Additionally, the "cost" of turning on and off the nuclear reactor is quite high, as this cost encompasses the safety considerations associated operating a nuclear reactor. However, the beneficial qualities of nuclear power are seen to outweigh the drawbacks of the increased switching cost when compared to the lower efficiency of a coal plant.

We first discuss 3, which displays the optimal switching strategies for a given current mode at day eighty-five of the ninety-day optimization period. Note that in general, it is not profitable to incur a loss in revenue by changing the operating mode of the plant, so most of the region remains the color which corresponds to the current mode. However, if switches do occur, they favor mode 2, where electricity is produced by a combination of nuclear and gas, and occur in regions where either the cost of nuclear is very low or the cost of the other fuels is very high. This occurrence is in line with the formulation of the problem, where nuclear

and gas power perform better than coal power. Empirically, therefore, the switching strategies provided by the algorithm appear reasonable.

Furthermore, as n decreases (meaning that there is more time left for the operator to make switching decisions before the end of the operational period at $n = N = 90$), switching costs make up a smaller proportion of total costs. We look at 4, which shows the optimal switching strategies at day thirty – much earlier in the process. At this point, there is also more uncertainty regarding the evolution of the state process, and this appears to correlate with the algorithm favoring the mode with the best "average" performance, which is mode 2, where electricity is produced by a combination of natural gas and nuclear power. Overwhelmingly, the region for which it is optimal to switch to mode 2 increases as n decreases, as shown in fig. 4.

Note that in fig. 4 the algorithm does not prescribe a switch when gas prices are high and when operating fully using gas. This may seem counterintuitive, as higher gas prices in comparison to other fuel prices make gas less attractive at first glance. However, these higher gas prices also result in much higher switching costs for the plant, as the switching costs are also directly proportional to the fuel costs in our model. This introduces an element of reluctance if the profit to be made from switching is small enough in comparison to expensive switching costs. In addition, the mean reversion of the gas prices implies downward pressure on high gas prices (in expectation). It can be seen that as the time horizon becomes longer this effect is weaker, but it still has some impact in fig. 4.

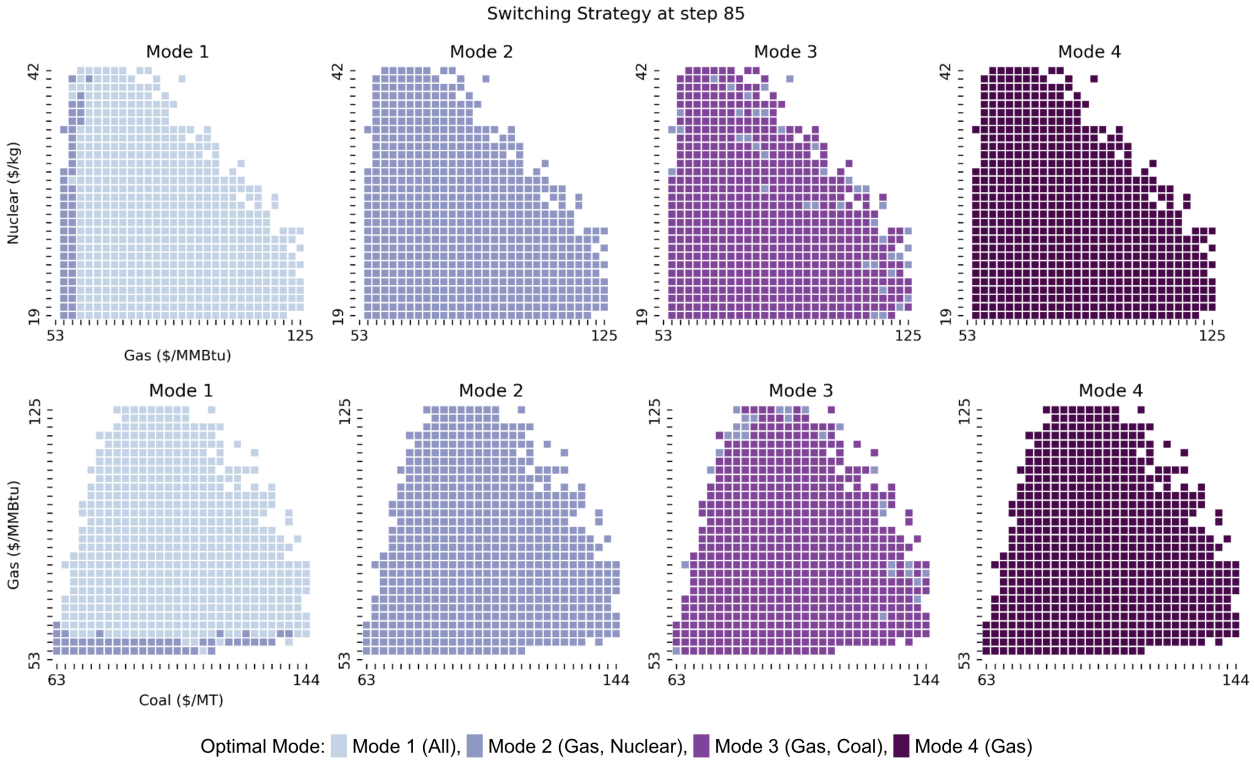

Figure 3. Switching strategies at $n = 85$, where n is the number of days that has elapsed since the start of the period and the period has length $N = 90$ days.

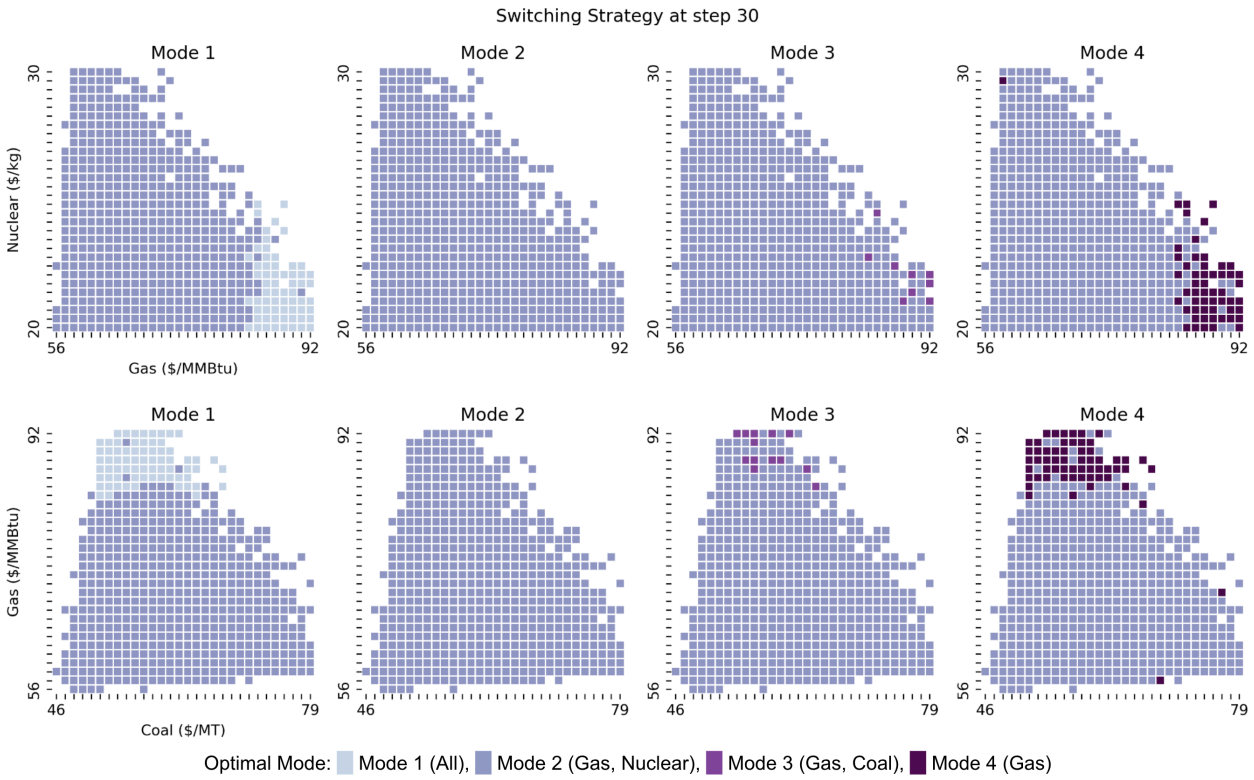

Figure 4. Switching strategies at $n = 30$, where n is the number of days that has elapsed since the start of the period and the period has length $N = 90$ days.

5. Convergence Analysis

Recall that we wish to prove that the error of the OSJ algorithm will converge to zero as the neural network approximation errors of the architecture converge to zero and as the size of the time steps, $\Delta t = T/M$, decreases to zero. The error in both theorem 1 and theorem 2 can be split into two parts: a discretization error for the stochastic processes and an algorithm-associated error. Once we have established the discretization error, we will be able to calculate the error of the neural network approximation by working with only discrete processes.

5.1. **Discretization Error.** As stated in remark 2, the result of theorem 1 follows when the discretization scheme used for simulating paths of X_t is of at least strong order 0.5 and the result of theorem 2 follows when the discretization scheme is of at least strong order 1.0 (the choice of scheme is arbitrary). For consistency, we explicitly define the discretization error as

$$\left(\mathbb{E}\left[\max_{n=1,\dots,N-1}|X_{t_{n+1}}-X^{\pi}_{n+1}|^2+\sup_{t\in[t_n,t_{n+1}]}|X_t-X^{\pi}_n|^2\right]\right)^{1/2}\leq C_X(\Delta t)^{\gamma}=O(M^{-\gamma})$$

when the discretization scheme is strong order γ. For our purposes, $\gamma = 0.5$ in section 5.2.1 and section 5.2.2 and $\gamma = 1.0$ in section 5.3.

Let us now move on to the error between $Y^i_t = V(t, X_t, i)$ and the discrete approximation of this process, $Y^{\pi,i}_n$, which approximates Y^i_t at time t_n starting in mode i. At the terminal time T, the terminal condition dictates that

$$Y^{\pi,i}_M \equiv g^i(X^{\pi}_M).$$

Earlier values of $Y^{\pi,i}_n$ for $n \in \{0, \dots, M-1\}$ can now be defined recursively in terms of the discrete approximation of the continuation values at each time step, represented by $\tilde{Y}^{\pi,i}_n$, following the relationship

$$Y^{\pi,i}_n := \mathbf{1}_{t_n\in\mathfrak{R}}\max\left\{\tilde{Y}^{\pi,i}_n,\ \max_{j\neq i}\left(-C_{i,j}(X^{\pi}_n)+\tilde{Y}^{\pi,j}_n\right)\right\}+\mathbf{1}_{t_n\notin\mathfrak{R}}\tilde{Y}^{\pi,i}_n,\forall i\in\mathbb{I},\forall n\in\{0,\dots,M-1\}. \tag{5.1}$$

Recall that $\mathfrak{R}$ is the switching grid, and the grid size is of order $O(M^{-1/2})$. This representation is driven by the dynamic programming principle. Therefore, we are able to solve for $Y_n^{\pi,i}$ by comparing the continuations values $\tilde{Y}_n^{\pi,i}$ at each step n. In turn, the continuation values at time t_n rely on the approximated value functions $Y_{n+1}^{\pi,i}$ from the previous step. The exact formulation of the discrete approximation of these continuation values, as well as discrete approximations of Z_t^i and ΔY_t^i, follows the method laid out in [8]. We set

$$\tilde{Y}_n^{\pi,i} := \mathbb{E}[Y_{n+1}^{\pi,i}|\mathcal{F}_n] + f(t_n, X_n^\pi)\Delta t, \tag{5.2}$$

$$Z_n^{\pi,i} := \frac{1}{\Delta t}\mathbb{E}[Y_{n+1}^{\pi,i}\Delta W_n|\mathcal{F}_n], \tag{5.3}$$

$$\Delta Y_n^{\pi,i} := \frac{1}{\lambda\Delta t}\mathbb{E}[Y_{n+1}^{\pi,i}\Delta \tilde{N}_n|\mathcal{F}_n]. \tag{5.4}$$

The strong order γ approximation of X_t where $\gamma \geq 0.5$ and the conditions in assumption 1 ensure that the discrete-time approximation scheme described in eq. (5.2)-eq. (5.4) is of strong order at least 0.5, as stated in Theorem 2.1 of [8], when considering the un-reflected BSDEs with jumps. Theorem 5.4 of the reference [13] presents the convergence rate for a similar discretization scheme for a multidimensional reflected BSDE without jumps. By following a similar path to these two papers, one which handles a BSDE with jumps and one which handles a BSDE with reflections which arise from switching, we can obtain a convergence rate for the above discretization scheme. For all modes i in $\mathbb{I}$ and for ε arbitrarily small, we have

$$\begin{aligned}
\max_{n\in 0,1,\dots,M}\mathbb{E}[\tilde{Y}_{t_n}^i - \tilde{Y}_n^{\pi,i}|^2 + |Y_{t_n}^i - Y_n^{\pi,i}|^2] &\leq C^\varepsilon M^{-1+\varepsilon},\\
\mathbb{E}\left[\sum_{n=0}^{M-1}\int_{t_n}^{t_{n+1}}|Z_t^i - Z_n^{\pi,i}|^2 dt\right] &\leq C^\varepsilon M^{-1/2+\varepsilon},\\
\mathbb{E}\left[\sum_{n=0}^{M-1}\int_{t_n}^{t_{n+1}}\left|\int_{\mathbb{R}^d}\Delta Y_t^i(e)\nu(de) - \Delta Y_n^{\pi,i}\right|^2 ds\right] &\leq C^\varepsilon M^{-1/2+\varepsilon}.
\end{aligned} \tag{5.5}$$

Therefore, in following sections, we will focus on the error between the discrete approximations $(Y_n^{\pi,i}, Z_n^{\pi,i}, \Delta Y_n^{\pi,i})$ and the neural network approximations $(\hat{\mathcal{Y}}_n^i(X_n^\pi), \hat{\mathcal{Z}}_n^i(X_n^\pi), \widehat{\Delta Y}_n^i(X_n^\pi))$.

Remark 8. *Each $\tilde{Y}_n^{\pi,i}$ can be represented as a conditional expectation with respect to $\mathcal{F}_n$, so due to the Markovian nature of the problem, it can be expressed as a function of $X_n^{\pi,i}$. We abuse notation and denote the value function by $\tilde{Y}_n^{\pi,i}(X_n^{\pi,i})$. The same can be done to represent $Y_n^{\pi,i}$ as $Y_n^{\pi,i}(X_n^\pi)$, $Z_n^{\pi,i}$ as $Z_n^{\pi,i}(X_n^\pi)$, and $\Delta Y_n^{\pi,i}$ as $\Delta Y_n^{\pi,i}(X_n^\pi)$.*

5.2. **Proof of theorem 1.**

5.2.1. *Value Function Learning Error.* We now want to bound the error between the function $\hat{\mathcal{Y}}_n^i(X_n^\pi)$ learned by the OSJ algorithm and the discrete value function approximation $Y_n^{\pi,i}(X_n^\pi)$, for any given starting mode i. Using $|\max\{a,b\} - \max\{c,d\}| \leq \max\{|a-c|, |b-d|\}$ we can rewrite

$$\begin{aligned}
|Y_n^{\pi,i}(X_n^\pi) - \hat{\mathcal{Y}}_n^i(X_n^\pi)| =& |\max\{\tilde{Y}_n^{\pi,i}(X_n^\pi), \max_{j\neq i}(-C_{i,j}(X_n^\pi) + \tilde{Y}_n^{\pi,j}(X_n^\pi))\}\\
&- \max\{\tilde{\mathcal{Y}}_n^i(X_n^\pi), \max_{j\neq i}(-C_{i,j}(X_n^\pi) + \tilde{\mathcal{Y}}_n^j(X_n^\pi))\}|\\
\leq& \max\{|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|,\\
&\quad |\max_{j\neq i}(-C_{i,j}(X_n^\pi) + \tilde{Y}_n^{\pi,j}(X_n^\pi) + C_{i,j}(X_n^\pi) - \tilde{\mathcal{Y}}_n^j(X_n^\pi))|\}\\
=& \max\{|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|, |\max_{j\neq i}(\tilde{Y}_n^{\pi,j}(X_n^\pi) - \tilde{\mathcal{Y}}_n^j(X_n^\pi))|\}\\
\leq& \max_{j\in\mathbb{I}}|\tilde{Y}_n^{\pi,j}(X_n^\pi) - \tilde{\mathcal{Y}}_n^j(X_n^\pi)|.
\end{aligned}$$

Therefore, we conclude that

$$\mathbb{E}\big[\max_{i\in\mathbb{I}}|Y_n^{\pi,i}(X_n^\pi) - \hat{\mathcal{Y}}_n^i(X_n^\pi)|^2\big] \leq \mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|^2\big]. \tag{5.6}$$

In light of this, let us focus on the mode-wise maximum of the difference in the continuation values $\tilde{\mathcal{Y}}_n^i$ and $\tilde{Y}_n^{\pi,i}$ by using eq. (3.5) and eq. (5.2) to find that

$$\tilde{Y}_n^{\pi,i}(X_n^\pi) - \hat{y}_n^i(X_n^\pi) = \mathbb{E}\big[Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|\mathcal{F}_n\big].$$

Squaring and taking expectation over the mode-wise maximum, then applying Jensen's inequality and tower property,

$$\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\big] \leq \mathbb{E}\big[\max_{i\in\mathbb{I}}|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big]. \tag{5.7}$$

We can get a lower bound on the left-hand side by using $\max_{i\in\mathbb{I}}|a_i - b_i| \geq |\max_{i\in\mathbb{I}}|a_i| - \max_{i\in\mathbb{I}}|b_i||$, such that

$$\mathbb{E}\left[\max_{i\in\mathbb{I}}\big|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\right] \geq \mathbb{E}\Big[\big|\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)| - \max_{i\in\mathbb{I}}|\tilde{\mathcal{Y}}_n^j(X_n^\pi) - \hat{y}_n^j(X_n^\pi)|\big|^2\Big].$$

Now, we apply Young's inequality in the form $(a-b)^2 \geq (1-\Delta t)a^2 - \frac{1}{\Delta t}b^2$ to the right-hand side so that a lower bound is obtained of the form

$$\begin{aligned}
&\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\big] \\
\geq&(1-\Delta t)\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|^2\big] - \frac{1}{\Delta t}\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\big] \\
\geq&(1-\Delta t)\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|^2\big] - \frac{1}{\Delta t}\mathbb{E}\Big[\sum_{i\in\mathbb{I}}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\Big]
\end{aligned}$$

By combining this with eq. (5.7) we can write

$$\begin{aligned}
(1-\Delta t)\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|^2\big] \leq&\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\big] \\
&+ \frac{1}{\Delta t}\mathbb{E}\Big[\sum_{i\in\mathbb{I}}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2\Big] \\
\leq&\mathbb{E}\big[\max_{i\in\mathbb{I}}|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big] \\
&+ \frac{1}{\Delta t}\sum_{i\in\mathbb{I}}\mathbb{E}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2.
\end{aligned}$$

For Δt sufficiently small (less than 1), we have

$$\begin{aligned}
\mathbb{E}\big[\max_{i\in\mathbb{I}}|\tilde{Y}_n^{\pi,i}(X_n^\pi) - \tilde{\mathcal{Y}}_n^i(X_n^\pi)|^2\big] \leq&(1+C_1\Delta t)\mathbb{E}\big[\max_{i\in\mathbb{I}}|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big] \\
&+ C_1 M\sum_{i\in\mathbb{I}}\mathbb{E}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2,
\end{aligned} \tag{5.8}$$

where C_1 does not grow as M increases and is independent of structure of the neural networks.

At this point, we have established an upper bound on the error between the discretized continuation values $\tilde{Y}_n^{\pi,i}$ and the neural-network-generated continuation values $\tilde{\mathcal{Y}}_n^i(X_n^\pi)$. This upper bound is in terms of $\mathbb{E}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2$ and $\mathbb{E}|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2$. However, theorem 1 states the convergence error in terms of the neural network approximation errors ε_n^y, ε_n^z, and ε_n^u. Therefore, our next course of action is to determine a bound on $\mathbb{E}|\tilde{\mathcal{Y}}_n^i(X_n^\pi) - \hat{y}_n^i(X_n^\pi)|^2$ in terms of these neural network approximation errors.

To do this, we must analyze the loss function described in eq. (3.3). Replace $\hat{\mathcal{Y}}^i_{n+1}(X^\pi_{n+1})$ in eq. (3.3) with its corresponding BSDE representation eq. (3.8) and use the relations eq. (3.9) and eq. (3.10) to write

$$\begin{aligned}
L^i_n(\theta) &= \mathbb{E}\Bigg|\hat{y}^i_n(X^\pi_n) - \mathcal{Y}^i_n(X^\pi_n,\theta) + \big(f_i(t_n,X^\pi_n) - f_i(t_n,X^\pi_n)\big)\Delta t \\
&\qquad + \int_{t_n}^{t_{n+1}}\int_{\mathbb{R}^d} u^i_s(e)\tilde{\mathcal{N}}(de,ds) - \Delta\mathcal{Y}^i_n(X^\pi_n,\theta)\Delta\tilde{N}_n + \int_{t_n}^{t_{n+1}}(z^i_s)^T dW_s - \mathcal{Z}^i_n(X^\pi_n,\theta)^T\Delta W_n\Bigg|^2 \\
&= \mathbb{E}\big|\hat{y}^i_n(X^\pi_n) - \mathcal{Y}^i_n(X^\pi_n,\theta) + \big(\hat{u}^i_n(X^\pi_n) - \Delta\mathcal{Y}^i_n(X^\pi_n,\theta)\big)\Delta\tilde{N}_n + \big(\hat{z}^i_n(X^\pi_n) - \mathcal{Z}^i_n(X^\pi_n,\theta)\big)^T\Delta W_n\big|^2 \\
&\quad + \mathbb{E}\left[\int_{t_n}^{t_{n+1}}\int_{\mathbb{R}^d}|u^i_s - \hat{u}^i_n(X^\pi_n)|^2\mathcal{N}(de,ds)\right] + \mathbb{E}\left[\int_{t_n}^{t_{n+1}}||z^i_s - \hat{z}^i_n(X^\pi_n)||^2 ds\right] \\
&= \bar{L}^i_n(\theta) + Error(U,Z),
\end{aligned}$$

where we used Itô isometry and the definitions of $\hat{u}^i_n$ and $\hat{z}^i_n$, given in eq. (3.7) and eq. (3.6) respectively, to split up the expectation. We introduce the notation

$$\bar{L}^i_n(\theta) := \mathbb{E}\big|\hat{y}^i_n(X^\pi_n) - \mathcal{Y}^i_n(X^\pi_n,\theta) + \big(\hat{u}^i_n(X^\pi_n) - \Delta\mathcal{Y}^i_n(X^\pi_n,\theta)\big)\Delta\tilde{N}_n + \big(\hat{z}^i_n(X^\pi_n) - \mathcal{Z}^i_n(X^\pi_n,\theta)\big)^T\Delta W_n\big|^2$$

and

$$Error(U,Z) := \mathbb{E}\left[\int_{t_n}^{t_{n+1}}\int_{\mathbb{R}^d}|u^i_s - \hat{u}^i_n(X^\pi_n)|^2\mathcal{N}(de,ds)\right] + \mathbb{E}\left[\int_{t_n}^{t_{n+1}}||z^i_s - \hat{z}^i_n(X^\pi_n)||^2 ds\right].$$

Note that $Error(U,Z)$ is independent of our choice of θ. In addition, this error scales with Δt as stated in [8], since $\hat{z}^i_n$ and $\hat{u}^i_n$ are the L^2 projections of z^i_s and u^i_s on $[t_n, t_{n+1})$. For the near future, we will focus on $\bar{L}^i_n(\theta)$. By Itô isometry and $\mathbb{E}[\Delta W_n] = 0$, $\mathbb{E}[\Delta\tilde{N}_n] = 0$, we obtain

$$\begin{aligned}
\bar{L}^i_n(\theta) =& \mathbb{E}|\hat{y}^i_n(X_n) - \mathcal{Y}^i_n(X_n,\theta)|^2 \\
&+ \Delta t\mathbb{E}||\hat{z}^i_n(X_n) - \mathcal{Z}^i_n(X_n,\theta)||^2 + \lambda\Delta t\mathbb{E}|\hat{u}^i_n(X_n) - \Delta\mathcal{Y}^i_n(X_n,\theta)|^2.
\end{aligned}$$

Extending this to the sum of the loss functions over all possible modes and choosing $\theta = (\theta_1,\theta_2,\theta_3) = \theta^{*,i}_n \in \arg\min_{\theta\in\Theta} L^i_n(\theta)$, we conclude that

$$\begin{aligned}
\sum_{i\in\mathbb{I}}\bar{L}^i_n(\theta^{*,i}_n) =& \sum_{i\in\mathbb{I}}\mathbb{E}|\hat{y}^i_n(X^\pi_n) - \tilde{\mathcal{Y}}^i_n(X^\pi_n)|^2 \\
&+ \sum_{i\in\mathbb{I}}\Delta t\big(\mathbb{E}||\hat{z}^i_n(X^\pi_n) - \hat{\mathcal{Z}}^i_n(X^\pi_n)||^2 + \lambda\mathbb{E}|\hat{u}^i_n(X^\pi_n) - \widehat{\Delta\mathcal{Y}}^i_n(X^\pi_n)|^2\big) \\
\le& \sum_{i\in\mathbb{I}}\big(\inf_{\theta_1}\mathbb{E}|\hat{y}^i_n(X^\pi_n) - \mathcal{Y}^i_n(X^\pi_n,\theta_1)|^2 + \Delta t\inf_{\theta_2}\mathbb{E}||\hat{z}^i_n(X^\pi_n) - \mathcal{Z}^i_n(X^\pi_n,\theta_2)||^2\big) \\
&+ \sum_{i\in\mathbb{I}}\lambda\Delta t\big(\inf_{\theta_3}\mathbb{E}|\hat{u}^i_n(X^\pi_n) - \Delta\mathcal{Y}^i_n(X^\pi_n,\theta_3)|^2\big) \\
=& \varepsilon^y_n + \Delta t(\varepsilon^z_n + \lambda\varepsilon^u_n).
\end{aligned} \tag{5.9}$$

We note that $\theta^{*,i}_n$ must minimize both $L^i_n(\theta)$ and $\bar{L}^i_n(\theta)$ because $Error(U,Z)$ does not depend on θ. This facilitates the above inequality, where the final line follows from the definition of the neural network errors given in eq. (3.11). From eq. (5.9), we also get the looser bound

$$\sum_{i\in\mathbb{I}}\mathbb{E}|\hat{y}^i_n(X^\pi_n) - \tilde{\mathcal{Y}}^i_n(X^\pi_n)|^2 \le C_2(\varepsilon^y_n + \Delta t(\varepsilon^z_n + \varepsilon^u_n)) = C_2\boldsymbol{\varepsilon^M_n}/M, \tag{5.10}$$

where $C_2 = \max\{\lambda, 1\}$. Applying this to eq. (5.8) and recalling eq. (5.6) yields

$$\begin{aligned}
\mathbb{E}\big[\max_{i\in\mathbb{I}}|Y^{\pi,i}_n(X^\pi_n) - \hat{\mathcal{Y}}^i_n(X^\pi_n)|^2\big] \le& (1 + C_1\Delta t)\mathbb{E}\big[\max_{i\in\mathbb{I}}|Y^{\pi,i}_{n+1}(X^\pi_{n+1}) - \hat{\mathcal{Y}}^i_{n+1}(X^\pi_{n+1})|^2\big] \\
&+ C_1C_2M(\boldsymbol{\varepsilon^M_n}/M).
\end{aligned} \tag{5.11}$$

This setup allows us to perform induction on the inequality, continuing until the right-hand side is expressed in terms of $\mathbb{E}\big[\max_{i\in\mathbb{I}}|Y^{\pi,i}_M(X^\pi_M) - \hat{\mathcal{Y}}^i_M(X^\pi_M)|^2\big] = 0$ and a sum of neural network errors at time steps from

n to M. Then, we can conclude the maximum error over the M time steps is bounded as

$$\max_{n=0,1,\dots,M-1} \mathbb{E}\Big[\max_{i\in\mathbb{I}} |Y_n^{\pi,i}(X_n^\pi) - \hat{\mathcal{Y}}_n^i(X_n^\pi)|^2\Big] \leq C_3 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M}. \tag{5.12}$$

where C_3 encompasses the coefficients C_1 and C_2, as well as coefficients arising from the inductive step. This aggregate coefficient, C_3, is independent of Δt and the neural network structure. Recall we initialized $\hat{\mathcal{Y}}_M^i(\cdot) = g^i(\cdot)$, so $\mathbb{E}[\max_{i\in\mathbb{I}} |Y_M^{\pi,i}(X_M^\pi) - \hat{\mathcal{Y}}_M^i(X_M^\pi)|^2] = 0$. However, the error in eq. (5.12) is only the error between the discrete approximation of the value function and the trained neural network approximation of the value function. The true error involves the continuous value function Y_t^i when starting in mode i at time t, and so the discrete approximation error given by eq. (5.5) must be incorporated into our final error bound. Therefore, the overall value function error is given by

$$\max_{n=0,1,\dots,M-1} \mathbb{E}\Big[\max_{i\in\mathbb{I}} |Y_{t_n}^i - \hat{\mathcal{Y}}_n^i(X_n^\pi)|^2\Big] \leq C_3 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M} + C^\varepsilon M^{-1/2+\varepsilon}. \tag{5.13}$$

5.2.2. *Auxiliary Function Learning Errors.* We must now verify that $Z_n^{\pi,i}$ and $\Delta Y_n^{\pi,i}$ are also approximated well. Looking first at the error in Z, we use triangle inequality to split each element of the sums into two parts and use eq. (5.9) on the summation to get

$$\begin{aligned}
\Delta t \max_{i\in\mathbb{I}} \mathbb{E}||Z_n^{\pi,i} - \hat{\mathcal{Z}}_n^i(X_n^\pi)||^2 \leq& 2\Delta t(\max_{i\in\mathbb{I}} \mathbb{E}||Z_n^{\pi,i} - \hat{z}_n^i(X_n^\pi)||^2] + \max_{i\in\mathbb{I}} \mathbb{E}||\hat{z}_n^i(X_n^\pi) - \hat{\mathcal{Z}}_n^i(X_n^\pi)||^2) \\
\leq& 2\Delta t \left(\max_{i\in\mathbb{I}} \mathbb{E}||Z_n^{\pi,i} - \hat{z}_n^i(X_n^\pi)||^2] + \sum_{i\in\mathbb{I}} \mathbb{E}||\hat{z}_n^i(X_n^\pi) - \hat{\mathcal{Z}}_n^i(X_n^\pi)||^2\right) \\
\leq& 2\Delta t \max_{i\in\mathbb{I}} \mathbb{E}||Z_n^{\pi,i} - \hat{z}_n^i(X_n^\pi)||^2 + C_2\big(\varepsilon_n^y + \Delta t\varepsilon_n^z + \Delta t\varepsilon_n^u\big).
\end{aligned}$$

Similarly for ΔY, we find that

$$\begin{aligned}
\Delta t \max_{i\in\mathbb{I}} \mathbb{E}|\Delta Y_n^{\pi,i} - \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi)|^2 \leq& 2\Delta t(\max_{i\in\mathbb{I}} \mathbb{E}|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2 + \max_{i\in\mathbb{I}} \mathbb{E}|\hat{u}_n^i(X_n^\pi) - \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi)|^2) \\
\leq& 2\Delta t \left(\max_{i\in\mathbb{I}} \mathbb{E}|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2 + \sum_{i\in\mathbb{I}} \mathbb{E}|\hat{u}_n^i(X_n^\pi) - \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi)|^2\right) \\
\leq& 2\Delta t \max_{i\in\mathbb{I}} \mathbb{E}|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2 + C_2\big(\varepsilon_n^y + \Delta t\varepsilon_n^z + \Delta t\varepsilon_n^u\big).
\end{aligned}$$

We now need to work with $\Delta t\mathbb{E}||Z_n^{\pi,i} - \hat{z}_n^i(X_n^\pi)||^2$ and $\Delta t\mathbb{E}|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2$. From the definitions of $\Delta Y_n^{\pi,i}$ and $\hat{u}_n^i$ and Cauchy–Schwartz of form $|\mathbb{E}[XY|A]|^2 \leq \mathbb{E}[X^2|A]\mathbb{E}[Y^2|A]$, we get

$$\begin{aligned}
|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2 =& \frac{1}{(\lambda\Delta t)^2}\big|\mathbb{E}[(Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi))\Delta\tilde{N}_n|\mathcal{F}_n]\big|^2 \\
\leq& \frac{1}{(\lambda\Delta t)^2}\mathbb{E}[|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big|\mathcal{F}_n]\mathbb{E}[(\Delta\tilde{N}_n)^2|\mathcal{F}_n] \\
=& \frac{1}{\lambda\Delta t}\mathbb{E}[|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big|\mathcal{F}_n].
\end{aligned}$$

Notice that again we used Itô isometry applied to $\mathbb{E}[(\Delta\tilde{N}_n)^2|\mathcal{F}_n]$. Therefore, it can be seen that

$$\begin{aligned}
\Delta t \max_{i\in\mathbb{I}} \mathbb{E}|\Delta Y_n^{\pi,i}(X_n^\pi) - \hat{u}_n^i(X_n^\pi)|^2 \leq& \frac{1}{\lambda}\max_{i\in\mathbb{I}} \mathbb{E}\Big[\mathbb{E}[|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2|\mathcal{F}_n]\Big] \\
=& \frac{1}{\lambda}\max_{i\in\mathbb{I}} \mathbb{E}|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2 \\
\leq& \frac{1}{\lambda}\mathbb{E}[\max_{i\in\mathbb{I}} |Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2].
\end{aligned}$$

Using a similar methodology and recalling that ΔW_n is d-dimensional and so $\mathbb{E}[||\Delta W_n||^2|\mathcal{F}_n] = d\Delta t$, we can also derive

$$\Delta t \max_{i\in\mathbb{I}} \mathbb{E}||Z_n^{\pi,i}(X_n^\pi) - \hat{z}_n^i(X_n^\pi)||^2 \leq d\mathbb{E}\big[\max_{i\in\mathbb{I}} \big|Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)\big|^2\big].$$

Let $\kappa = d + \frac{1}{\lambda}$, so we can write

$$\begin{aligned}&\Delta t \max_{i\in\mathbb{I}} \left(\mathbb{E}||Z_n^{\pi,i} - \hat{z}_n^i(X_n^\pi)||^2 + \mathbb{E}|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2\right)\\ \leq &\kappa \mathbb{E}\big[\max_{i\in\mathbb{I}} |Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big].\end{aligned}$$

Now sum over all M. We use $Y_M^{\pi,i}(x) \equiv \hat{\mathcal{Y}}_M^i(x) \equiv g^i(x)$ and induction on eq. (5.11) (similar to the calculations done to yield eq. (5.12)) to yield

$$\begin{aligned}&\sum_{n=0}^{M-1} \Delta t \max_{i\in\mathbb{I}} \left(\mathbb{E}||Z_n^{\pi,i}(X_n^\pi) - \hat{z}_n^i(X_n^\pi)||^2 + \mathbb{E}|\Delta Y_n^{\pi,i} - \hat{u}_n^i(X_n^\pi)|^2\right)\\ \leq &\sum_{n=0}^{M-1} \kappa \mathbb{E}\big[\max_{i\in\mathbb{I}} |Y_{n+1}^{\pi,i}(X_{n+1}^\pi) - \hat{\mathcal{Y}}_{n+1}^i(X_{n+1}^\pi)|^2\big]\\ = &\sum_{n=1}^{M-1} \kappa \mathbb{E}\big[\max_{i\in\mathbb{I}} |Y_n^{\pi,i}(X_n^\pi) - \hat{\mathcal{Y}}_n^i(X_n^\pi)|^2\big] + \kappa \mathbb{E}\big[\max_{i\in\mathbb{I}} |Y_M^{\pi,i}(X_M^\pi) - \hat{\mathcal{Y}}_M^i(X_M^\pi)|^2\big]\\ \leq &C_3\kappa \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M}.\end{aligned}$$

Therefore,

$$\begin{aligned}\sum_{n=0}^{M-1} \Delta t \max_{i\in\mathbb{I}} \mathbb{E}||Z_n^{\pi,i} - \hat{\mathcal{Z}}_n^i(X_n^\pi)||^2 \leq &C_4 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M},\\ \sum_{n=0}^{M-1} \Delta t \max_{i\in\mathbb{I}} \mathbb{E}|\Delta Y_n^{\pi,i} - \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi)|^2 \leq &C_4 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M},\end{aligned}$$

where $C_4 = 2C_3\kappa + C_2$ and is independent of Δt. Similarly to in the previous section, the error between the continuous quantities Z_t^i and ΔY_t^i and their neural network approximations is made up of the above quantities and the discrete approximation error eq. (5.5), so the final error for these auxiliary processes is given by

$$\begin{aligned}\sum_{n=0}^{M-1} \max_{i\in\mathbb{I}} \int_{t_n}^{t_{n+1}} \mathbb{E}||Z_t^i - \hat{\mathcal{Z}}_n^i(X_n^\pi)||^2 dt \leq &C_4 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M} + C^\varepsilon M^{-1/2+\varepsilon},\\ \sum_{n=0}^{M-1} \max_{i\in\mathbb{I}} \int_{t_n}^{t_{n+1}} \mathbb{E}|\Delta Y_t^i - \widehat{\Delta\mathcal{Y}}_n^i(X_n^\pi)|^2 dt \leq &C_4 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M} + C^\varepsilon M^{-1/2+\varepsilon}.\end{aligned}$$

5.3. **Proof of theorem 2.** We show that the expected payoff of the switching strategy generated by the neural network converges to that of the true optimal strategy.

Remark 9. *We introduce new notation for this subsection, which explicitly highlights the dependence of the neural network output on the number of time steps, M. Specifically we redefine $\hat{\mathcal{Y}}_n^i(X_n^\pi)$ as $\hat{\mathcal{Y}}_{n,M}^i(X_n^\pi)$ and $\hat{y}_n^i(X_n^\pi)$ as $\hat{y}_{n,M}^i(X_n^\pi)$.*

Recall that $\boldsymbol{a^{NN,M}}$ is the strategy produced by the OSJ algorithm when the number of time steps is given by M. By definition, $\boldsymbol{a^{NN,M}}$ is a discrete switching strategy where switches can only occur at times t_n for $n = 0, \dots, M$ and $\boldsymbol{a^{NN,M}}$ takes on the value $\alpha_n^{NN,M,i}$ (shortened hereafter to α_n) on the interval $[t_n, t_{n+1})$. If the system was in mode α_{n-1} right before time t_n, then at time t_n the neural network value function satisfies the relationship

$$\hat{\mathcal{Y}}_{n,M}^{\alpha_{n-1}}(X_n^\pi) = \tilde{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi). \tag{5.14}$$

Remark 10. *Note that it will often be the case that $\alpha_{n-1} = \alpha_n$, in which case $C_{\alpha_{n-1},\alpha_n} = 0$ by assumption 2 and no switching cost is incurred. We investigate this further in lemma 3.*

The OSJ-generated strategy produces an expected payoff $J(0, x_0, i, \boldsymbol{a}^{\boldsymbol{NN,M}})$ when starting in initial mode i and initial state x_0. Let us now consider the error at some arbitrary time t_n between the expected payoff using the neural network strategy and the true value function. For any $n = 0, 1, 2, \ldots, M$, we can use the triangle inequality to yield

$$|Y_{t_n}^{\alpha_n} - J(t_n, X_{t_n}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})| \leq |Y_{t_n}^{\alpha_n} - \hat{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi)| + |\hat{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - J(t_n, X_{t_n}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})|.$$

We can also apply eq. (5.14) to produce

$$\begin{aligned}\hat{\mathcal{Y}}_{n,M}^{\alpha_n} - J(t_n, X_{t_n}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}}) =& \tilde{\mathcal{Y}}_{n,M}^{\alpha_n} - C_{\alpha_{n-1},\alpha_n}(X_n^\pi) - J(t_n, X_{t_n}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}}) \\ =& (\tilde{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - \hat{y}_{n,M}^{\alpha_n}(X_n^\pi)) \\ &+ (\hat{y}_{n,M}^{\alpha_n}(X_n^\pi) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi) - J(t_n, X_{t_n}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})).\end{aligned}$$

We can use eq. (2.2) to express the expected payoff $J(t_n, X_{t_n}, \alpha_{n-1}, \boldsymbol{a}^{\boldsymbol{NN,M}})$ recursively as

$$\begin{aligned}J(t_n, X_{t_n}, \alpha_{n-1}, \boldsymbol{a}^{\boldsymbol{NN,M}}) = \mathbb{E}\Big[&\int_{t_n}^{t_{n+1}} f_{\alpha_n}(s, X_s)ds - C_{\alpha_{n-1},\alpha_n}(X_{t_n}) \\ &+ J(t_{n+1}, X_{t_{n+1}}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})\Big|\mathcal{F}_n\Big].\end{aligned}$$

Therefore, recalling the definition eq. (3.5) and taking the expectation of the absolute value of the difference between $\hat{\mathcal{Y}}_{n,M}^{\alpha_{n-1}}$ and $J(t_n, X_{t_n}, \alpha_{n-1}, \boldsymbol{a}^{\boldsymbol{NN,M}})$ yields

$$\begin{aligned}\mathbb{E}|\hat{\mathcal{Y}}_{n,M}^{\alpha_{n-1}} - J(t_n, X_{t_n}, \alpha_{n-1}, \boldsymbol{a}^{\boldsymbol{NN,M}})| \leq& \mathbb{E}|\tilde{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - \hat{y}_{n,M}^{\alpha_n}(X_n^\pi)| \\ &+ \mathbb{E}|\hat{\mathcal{Y}}_{n+1,M}^{\alpha_n}(X_{n+1}^\pi) - J(t_{n+1}, X_{t_{n+1}}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})| \\ &+ \mathbb{E}|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)| \\ &+ \mathbb{E}\Big[\int_{t_n}^{t_{n+1}} |f_{\alpha_n}(t_n, X_n^\pi) - f_{\alpha_n}(s, X_s)|ds\Big].\end{aligned}$$

Taking the sum over $n = 0, \ldots, M-1$ on both sides gives us

$$\begin{aligned}\sum_{n=0}^{M-1} \mathbb{E}|\hat{\mathcal{Y}}_{n,M}^{\alpha_{n-1}} - J(t_n, X_{t_n}, \alpha_{n-1}, \boldsymbol{a}^{\boldsymbol{NN,M}})| \leq& \sum_{n=0}^{M-1} \mathbb{E}|\tilde{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - \hat{y}_{n,M}^{\alpha_n}(X_n^\pi)| \\ &+ \sum_{n=0}^{M-1} \mathbb{E}|\hat{\mathcal{Y}}_{n+1,M}^{\alpha_n}(X_{n+1}^\pi) - J(t_{n+1}, X_{t_{n+1}}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})| \\ &+ \sum_{n=0}^{M-1} \mathbb{E}|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)| \\ &+ \sum_{n=0}^{M-1} \mathbb{E}\Big[\int_{t_n}^{t_{n+1}} |f_{\alpha_n}(t_n, X_n^\pi) - f_{\alpha_n}(s, X_s)|ds\Big].\end{aligned}$$

By subtracting $\sum_{n=1}^{M-1} \mathbb{E}|\hat{\mathcal{Y}}_{n,M}^{\alpha_n} - J(t_n, X_{t_n}, \alpha_n, \boldsymbol{a}^{\boldsymbol{NN,M}})|$ from both sides, we obtain

$$\begin{aligned}|\hat{\mathcal{Y}}_{0,M}^{i}(x_0) - J(0, x_0, i, \boldsymbol{a}^{\boldsymbol{NN,M}})| \leq& \sum_{n=0}^{M-1} \mathbb{E}|\tilde{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - \hat{y}_{n,M}^{\alpha_n}(X_n^\pi)| \\ &+ \mathbb{E}|\hat{\mathcal{Y}}_{M,M}^{\alpha_{M-1}}(X_M^\pi) - J(T, X_T, \alpha_{M-1}, \boldsymbol{a}^{\boldsymbol{NN,M}})| \\ &+ \sum_{n=0}^{M-1} \mathbb{E}|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)| \\ &+ \sum_{n=0}^{M-1} \mathbb{E}\Big[\int_{t_n}^{t_{n+1}} |f_{\alpha_n}(t_n, X_n^\pi) - f_{\alpha_n}(s, X_s)|ds\Big].\end{aligned}$$

We make a few comments at this stage. First, at the terminal time $T = t_M$, $\hat{\mathcal{Y}}_{M,M}^{i}(x) \equiv J(T, x, i, \cdot) \equiv g^i(x), \forall x \in \mathbb{R}^d, i \in \mathbb{I}$. In addition, g^i is Lipschitz for all $i \in \mathbb{I}$. Therefore, the error at the final step M has

error at most $O(M^{-1})$ from the error between the discrete approximation X_M^π and the true random variable X_T. Second, notice that $|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)| \neq 0$ if and only if $\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}} = 1$. This property, along with the inequality $||\mathbf{x}||_1 \leq \sqrt{M}||\mathbf{x}||_2, \forall \mathbf{x} \in \mathbb{R}^M$, gives us

$$\begin{aligned}
|\hat{\mathcal{Y}}^i_{0,M}(x_0) - J(0,x_0,i,\boldsymbol{a}^{\boldsymbol{NN,M}})| \leq & \sum_{n=0}^{M-1} \mathbb{E}|\tilde{\mathcal{Y}}^{\alpha_n}_{n,M}(X_n^\pi) - \hat{y}^{\alpha_n}_{n,M}(X_n^\pi)| + O(M^{-1}) \\
& + \sum_{n=0}^{M-1} \mathbb{E}\big[|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)|\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}\big] \\
& + \sum_{n=0}^{M-1} \mathbb{E}\Big[\int_{t_n}^{t_{n+1}} |f_{\alpha_n}(t_n,X_n^\pi) - f_{\alpha_n}(s,X_s)|ds\Big] \\
& \leq \sum_{n=0}^{M-1} \mathbb{E}|\tilde{\mathcal{Y}}^{\alpha_n}_{n,M}(X_n^\pi) - \hat{y}^{\alpha_n}_{n,M}(X_n^\pi)| + O(M^{-1}) \\
& + \Big[M \sum_{n=0}^{M-1} \big(\mathbb{E}\big[|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)|\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}\big]\big)^2\Big]^{1/2} \\
& + \sum_{n=0}^{M-1} \mathbb{E}\Big[\int_{t_n}^{t_{n+1}} |f_{\alpha_n}(t_n,X_n^\pi) - f_{\alpha_n}(s,X_s)|ds\Big].
\end{aligned}$$

To proceed, we recall the Lipschitz assumption on the running cost f given by assumption 3 and the assumption of a strong order 1.0 discrete approximation of X_t, as well as the result eq. (5.10). Then, the inequality can be further simplified as

$$\begin{aligned}
|\hat{\mathcal{Y}}^i_{0,M}(x_0) - J(0,x_0,i,\boldsymbol{a}^{\boldsymbol{NN,M}})| \leq & \sum_{n=0}^{M-1} \big(C_2\boldsymbol{\varepsilon}^{\boldsymbol{M}}_{\boldsymbol{n}}/M\big) + O(M^{-1}) \\
& + \left[M \sum_{n=0}^{M-1} \big(\mathbb{E}\big[|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)|\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}\big]\big)^2\right]^{1/2}.
\end{aligned}$$

Focusing on the sum of the switching costs, we apply the Lipschitz assumption for the switching costs given by assumption 2, Cauchy–Schwartz inequality, and remark 2 to yield

$$\begin{aligned}
& \sum_{n=0}^{M-1} \big(\mathbb{E}\big[|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)|\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}\big]\big)^2 \\
\leq & [C]_l^2 \sum_{n=0}^{M-1} \big(\mathbb{E}\big[||X_{t_n} - X_n^\pi||\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}\big]\big)^2 \\
\leq & [C]_l^2 \sum_{n=0}^{M-1} \mathbb{E}\big[||X_{t_n} - X_n^\pi||^2\big]\mathbb{E}[(\mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}})^2] \\
\leq & O(M^{-2}) \times \mathbb{E}\left[\sum_{n=0}^{M-1} \mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}\right]. \qquad (5.15)
\end{aligned}$$

We define a new random variable S_M, representing the number of switches following from a given learned strategy neural network strategy $\boldsymbol{a}^{\boldsymbol{NN,M}}$. We define it as

$$S_M := \sum_{n=0}^{M-1} \mathbf{1}_{\{\alpha_{n-1}\neq\alpha_n\}}, \forall M \in \mathbb{N},$$

where $\{\alpha_n\}_{n=0}^{M-1}$ are determined according to $\boldsymbol{a}^{\boldsymbol{NN,M}}$ and the choice of $\alpha_{-1} = i \in \mathbb{I}$.

Lemma 3. *The expected number of switches incurred by* $\boldsymbol{a^{NN,M}}$ *is bounded by*

$$\mathbb{E}[S_M] \leq O(1) + C_5 \sum_{n=0}^{M-1} \left[\left(\boldsymbol{\varepsilon_n^M}/M \right)^{1/2} + \boldsymbol{\varepsilon_n^M} \right],$$

where $C_5 = C_2 + dC_3$.

Proof. From theorem 1 and specifically eq. (5.13), we have shown that $\hat{\mathcal{Y}}_0^i(x_0)$ should converge to Y_0^i for all $i \in I$ as $M \to \infty$. We will use this to prove that a bound exists for $\mathbb{E}[S_M]$. Recall that

$$\hat{\mathcal{Y}}_{n,M}^{\alpha_{n-1}}(X_n^\pi) = \mathbb{E}\left[f_{\alpha_n}(X_n^\pi)\frac{1}{M} - C_{\alpha_{n-1},\alpha_n}(X_n^\pi) + \hat{\mathcal{Y}}_{n+1,M}^{\alpha_n}(X_{n+1}^\pi) + \tilde{\mathcal{Y}}_{n,M}^{\alpha_n}(X_n^\pi) - \hat{y}_{n,M}^i(X_n^\pi) | \mathcal{F}_n \right].$$

Taking expectation with respect to $\mathcal{F}_0$, summing over $n \in \{0, \dots, M-1\}$, and rearranging yields

$$\sum_{n=0}^{M-1} \mathbb{E}[C_{\alpha_{n-1},\alpha_n}(X_n^\pi)] = \frac{1}{M}\sum_{n=0}^{M-1} \mathbb{E}[f_{\alpha_n}(X_n^\pi)] - \sum_{n=0}^{M-1} \mathbb{E}[\tilde{\mathcal{Y}}_{n,M}^i(X_n^\pi) - \hat{y}_{n,M}^i(X_n^\pi)] + g^i(X_M^\pi) - \hat{\mathcal{Y}}_{0,M}^i(x_0).$$

Therefore, from triangle inequality,

$$\Big| \sum_{n=0}^{M-1} \mathbb{E}[C_{\alpha_{n-1},\alpha_n}(X_n^\pi)] \Big| \leq \sum_{n=0}^{M-1} \frac{\mathbb{E}|f_{\alpha_n}(X_n^\pi)|}{M} + \sum_{n=0}^{M-1} \mathbb{E}|\tilde{\mathcal{Y}}_{n,M}^i(X_n^\pi) - \hat{y}_{n,M}^i(X_n^\pi)| + |\hat{\mathcal{Y}}_{0,M}^i(x_0)| + |g^i(X_M^\pi)|.$$

Note that switching costs are always positive if $i \neq j$, and specifically $C_{i,j}(x) \geq \epsilon > 0, \forall i \neq j \in \mathbb{I}$ from assumption 2. This implies that

$$\Big| \sum_{n=0}^{M-1} \mathbb{E}[C_{\alpha_{n-1},\alpha_n}(X_n^\pi)] \Big| \geq \sum_{n=0}^{M-1} \epsilon \mathbb{E}[\mathbf{1}_{\{\alpha_{n-1} \neq \alpha_n\}}] \geq \epsilon \mathbb{E}[S_M].$$

Now it remains to show that this quantity is bounded above by a quantity which converges to zero. Observe $|g^i(X_M^\pi)| < \infty$. We first tackle the boundedness of $|\hat{\mathcal{Y}}_{0,M}^i(x_0)|$. The solution to the original optimization problem, $V(0, x_0, i)$, always has a finite-valued expectation, meaning that $\mathbb{E}|Y_t^i| < \infty$ for all times $t \in [0, T]$ (though we are only looking at $t = 0$ at the moment) and all modes $i \in \mathbb{I}$. Therefore, using eq. (5.13),

$$\begin{aligned}
\mathbb{E}|\hat{\mathcal{Y}}_{0,M}^i(x_0)| \leq& \mathbb{E}|\hat{\mathcal{Y}}_{0,M}^i(x_0) - Y_0^i| + \mathbb{E}|Y_0^i| \leq d\mathbb{E}|\hat{\mathcal{Y}}_{0,M}^i(x_0) - Y_0^i|^2 + \mathbb{E}|Y_0^i| \\
\leq& dC_3 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M} + O(M^{-1}) + \mathbb{E}|Y_0^i|.
\end{aligned}$$

We can directly use assumption 3 to bound $\mathbb{E}|f_{\alpha_n}(X_n^\pi)|$. We also note that $\mathbb{E}|X| \leq (\mathbb{E}[|X|^2])^{1/2}$ and apply eq. (5.10) to bound $\mathbb{E}|\tilde{\mathcal{Y}}_{n,M}^i(X_n^\pi) - \hat{y}_{n,M}^i(X_n^\pi)|^2 \leq C_2 \boldsymbol{\varepsilon_n^M}/M$, giving us

$$\epsilon \mathbb{E}[S_M] \leq O(M^{-1/2}) + \sum_{n=0}^{M-1} \left(C_2 \boldsymbol{\varepsilon_n^M}/M \right)^{1/2} + dC_3 \sum_{n=0}^{M-1} \boldsymbol{\varepsilon_n^M} + \max_{i \in \mathbb{I}} \mathbb{E}|Y_0^i|.$$

To obtain the bound stated in the lemma, we set $C_5 = (\sqrt{C_2} + dC_3)/\epsilon$. We also note that $O(1)$ dominates $O(M^{-1/2})$, and this concludes the proof. □

We can now use this bound to continue investigating the error accrued by the neural network-produced switching strategy. We return to eq. (5.15) and, armed with lemma 3, conclude that the switching costs can be bounded as

$$\begin{aligned}
&\sum_{n=0}^{M-1} \mathbb{E}\big[|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)| \big] \\
\leq& M^{1/2} \left(\sum_{n=0}^{M-1} \left(\mathbb{E}\big[|C_{\alpha_{n-1},\alpha_n}(X_{t_n}) - C_{\alpha_{n-1},\alpha_n}(X_n^\pi)| \mathbf{1}_{\{\alpha_{n-1} \neq \alpha_n\}} \big] \right)^2 \right)^{1/2} \\
\leq& O(M^{-1/2}) \left(O(1) + C_5 \sum_{n=0}^{M-1} \left[\left(\boldsymbol{\varepsilon_n^M}/M \right)^{1/2} + \boldsymbol{\varepsilon_n^M} \right] \right)^{1/2}.
\end{aligned}$$

Therefore, returning to our analysis of the error between the neural network value function and the expected payoff, we incorporate the other error elements to yield

$$|\hat{\mathcal{Y}}^i_{0,M}(x_0) - J(0,x_0,i,\boldsymbol{a^{NN,M}})| \leq C_2 \sum_{n=0}^{M-1} \frac{\boldsymbol{\varepsilon^M_n}}{M} + O\left(\frac{1}{M^2}\right) \sqrt{O(1) + C_5 \sum_{n=0}^{M-1} \left[\left(\boldsymbol{\varepsilon^M_n}/M\right)^{1/2} + \boldsymbol{\varepsilon^M_n} \right]}.$$

We now must add $|Y^i_0 - \hat{\mathcal{Y}}^i_{0,M}(x_0)|$ which is bounded by eq. (5.13) and take the mode-wise maximum over i to achieve our final error bound

$$\begin{aligned}\max_{i\in\mathbb{I}} |V(0,x_0,i) - J(0,x_0,i,\boldsymbol{a^{NN,M}})| \leq & C^{\varepsilon} M^{-1/2+\varepsilon} + C_2 \sum_{n=0}^{M-1} \left[\frac{\boldsymbol{\varepsilon^M_n}}{M} + \boldsymbol{\varepsilon^M_n} \right] \\ & + O\left(\frac{1}{M^2}\right) \sqrt{O(1) + C_5 \sum_{n=0}^{M-1} \left[\left(\boldsymbol{\varepsilon^M_n}/M\right)^{1/2} + \boldsymbol{\varepsilon^M_n} \right]}.\end{aligned}$$

which is consistent with eq. (3.13).

6. Conclusion

We have developed an algorithm which uses neural networks and the dynamic programming principle to apply a backward-in-time approach to optimal switching problems in high dimensions. The algorithm is also novel in its ability to accommodate high-dimensional state processes with a finite-variational jump component. We have both applied our approach successfully to numerical examples and obtained analytical convergence results. Our preliminary numerical results indicate that our algorithm is able to accommodate state processes with finite-variational jumps when solving optimal switching problems, and specifically problems related to energy markets. The analytical results present the convergence in terms of the neural network approximation errors of the networks, which have been previously proven to converge to zero.

Tests in lower dimensions where comparisons can be made against probabilistic methods have verified the accuracy of the algorithm, and tests in higher dimensions that are intractable with more traditional approaches have demonstrated that computation time decreases sub-linearly with dimension. Therefore, this algorithm, while not as fast as other methods for low (i.e. two-dimensional) problems, is an excellent candidate for solving high-dimensional optimal switching problems with jumps, as it is impressively robust with respect to the typical slowdowns experienced as a result of the curse of dimensionality.

In future, we would like to apply our algorithm to a larger class of optimization problems. There are many problems in energy markets and in other real-world applications in which the state variable is control-dependent. If this is the case, then the chosen switching strategy will affect the evolution of other quantities. For example, the price of electricity in one period (which depends on the chosen level of electricity production) might affect demand in the next period. This would reflect consumer behavioral trends, but would introduce new computational challenges. Namely, it would not be possible to simulate paths for the state process before the optimization was calculated, and so the backward-in-time nature of the OSJ algorithm would not be suited to such problems and a new approach would be necessary.

While renewable energy sources like wind and solar energy do not incur carbon dioxide emission penalties and are technically free to obtain once installed, they still incur costs to bring online and take offline, and experience much more volatility in their realized capacity. Therefore, adding these sources also adds an increased risk of electricity underproduction. In the future, we could develop an algorithm which contains a realistic method of penalizing unmet electricity demand, and therefore allows us to investigate interesting problems involving "free", renewable energy sources. Other potential new models could include more complex scheduling options (essentially allowing the electricity producer to consider a greater number of modes when scheduling).

Another direction could be investigating or improving the robustness of these algorithms. Robustness is crucial in energy applications because electricity producers wish to avoid "worst-case" scenarios, such as demand spikes or failures in the power grid. The solution to a given class of robust switching control problems is characterized in [4], and expanded to infinite time-horizon ergodic problems in [3]. It could be productive to expand this theoretical framework by designing a neural network-based algorithm for such robust optimal switching problems.

Finally, new convergence results for neural networks are being developed constantly. Another useful extension could be to present convergence results in terms of concrete quantities such as the neural network width, depth, or overall size. Such results have been calculated for specific classes of neural networks, as in [41, 38]

Acknowledgments

We would like to thank the referees and associate editor for their time, energy, and valuable comments which greatly improved the quality of our paper.

Appendix A. Parameters for Numerical Experiments

A.1. **Parameters used in section 4.2.** Below are the parameters used for the dynamics of $X_t = [D_t, A_t^1, A_t^2, A_t^3, S_t^0, S_t^1, S_t^2, S$ $\mathbb{R}^9$, as well as for the running costs f and switching costs $C_{i,j}$ for section 4.2.

Parameter	Value
α	$[4, 8, 8, 8]^T$
β	$\begin{bmatrix} 15 & 0.1 & 0.1 & 0 \\ 0.1 & 0.5 & -0.1 & 0 \\ 0.1 & -0.1 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$
X_0	$[0, 0, 0, 0, 20, 40, 60, 20, 120]^T$
μ	$\begin{bmatrix} -4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & -1 \end{bmatrix}$
Σ	$\frac{1}{100}\begin{bmatrix} 2.5 & 1.25 & 1.25 & 1.25 & 1.25 \\ 1.25 & 5 & 1.25 & 1.25 & 1.25 \\ 1.25 & 1.25 & 15 & 1.25 & 1.25 \\ 0.25 & 0.25 & 0.25 & 1.5 & 1.25 \\ 1.25 & 1.25 & 1.25 & 1.25 & 3 \end{bmatrix}$
h^0	$[0.5, 2, 0]^T$
h	$[1, 1.5, 1.5]^T$
$\boldsymbol{c}$	$[0.1, 0.1, 0.5]^T$
ϵ	0.001

Table 5. Parameters associated with section 4.2.

References

[1] R. Aïd, L. Campi, N. Langrené, and H. Pham. A Probabilistic Numerical Method for Optimal Multiple Switching Problems in High Dimension. *SIAM J. Financial Math.*, 5:191–231, 01 2014.

[2] C. Barrera-Esteve, F. Bergeret, C. H. Dossal, E. Gobet, A. Meziou, R. Munos, and D. Reboul-Salze. Numerical methods for the pricing of Swing options: a stochastic control approach. *Methodol. Comput. Appl.*, 8(4):517–540, Dec. 2006.

[3] E. Bayraktar, A. Cosso, and H. Pham. Robust Feedback Switching Control: Dynamic Programming and Viscosity Solutions. *SIAM J Control Optim*, 54(5):2594–2628, 2016.

[4] E. Bayraktar, A. Cosso, and H. Pham. Ergodicity of Robust Switching Control and Nonlinear System of Quasi-Variational Inequalities. *SIAM J Control Optim*, 55(3):1915–1953, 2017.

[5] E. Bayraktar and M. Egami. On the One-Dimensional Optimal Switching Problem. *Math. Oper. Res.*, 35(1):140–159, 2010.

[6] S. Becker, P. Cheridito, and A. Jentzen. Pricing and Hedging American-Style Options with Deep Learning. *J. Risk Financ. Manag.*, 13(7), 2020.

[7] I. H. Biswas, E. R. Jakobsen, and K. H. Karlsen. Viscosity Solutions for a System of Integro-PDEs and Connections to Optimal Switching and Control of Jump-Diffusion Processes. *Appl Math Optim*, 62(1):47–80, 2010.

[8] B. Bouchard and R. Elie. Discrete-time approximation of decoupled Forward–Backward SDE with jumps. *Stochastic Process. Appl.*, 118(1):53 – 75, 2008.

[9] B. Bouchard and N. Touzi. Discrete-time approximation and monte-carlo simulation of backward stochastic differential equations. *Stochastic Process. Appl.*, 111(2):175–206, 2004.

[10] N. Bruti-Liberati and E. Platen. Strong approximations of stochastic differential equations with jumps. *Journal of Computational and Applied Mathematics*, 205(2):982–1001, 2007. Special issue on evolutionary problems.

[11] N. Bruti-Liberati and E. Platen. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[12] R. Carmona and M. Ludkovski. Pricing Asset Scheduling Flexibility using Optimal Switching. *Appl. Math. Finance*, 15(5-6):405–447, 2008.

[13] J.-F. Chassagneux, R. Elie, and I. Kharroubi. Discrete-time approximation of multidimensional BSDEs with oblique reflections. *Ann. Appl. Probab.*, 22(3):971 – 1007, 2012.

[14] J.-F. Chassagneux and A. Richou. Rate of convergence for the discrete-time approximation of reflected BSDEs arising in switching problems. *Stochastic Process. Appl.*, 129(11):4597 – 4637, 2019.

[15] N. El Karoui, C. Kapoudjian, E. Pardoux, S. Peng, and M. C. Quenez. Reflected solutions of backward SDE's, and related obstacle problems for PDE's. *Ann. Probab.*, 25(2):702 – 737, 1997.

[16] R. Elie and I. Kharroubi. BSDE representations for optimal switching problems with controlled volatility. *Stochastics and Dynamics*, 14:1450003, 2014.

[17] E. Feinberg and X. Zhang. Optimizing Cloud Utilization via Switching Decisions. *SIGMETRICS Perform. Eval. Rev.*, 41(4):57–60, apr 2014.

[18] B. J. Felix and C. Weber. Gas storage valuation applying numerically constructed recombining trees. *European J. Oper. Res.*, 216(1):178–187, 2012.

[19] R. Frey and V. Köck. Deep neural network algorithms for parabolic pides and applications in insurance mathematics. In M. Corazza, C. Perna, C. Pizzi, and M. Sibillo, editors, *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, pages 272–277, Cham, 2022. Springer International Publishing.

[20] M. Germain, H. Pham, and X. Warin. Approximation Error Analysis of Some Deep Backward Schemes for Nonlinear PDEs. *SIAM J. Sci. Comput.*, 44(1):A28–A56, 2022.

[21] A. Gnoatto, M. Patacca, and A. Picarelli. A Deep Solver for BSDEs with Jumps. 2022.

[22] S. Hamadène and M. Jeanblanc. On the Starting and Stopping Problem: Application in Reversible Investments. *Math. Oper. Res.*, 32(1):182–192, 2007.

[23] S. Hamadène and J. Zhang. Switching problem and related system of reflected backward SDEs. *Stochastic Processes and their Applications*, 120(4):403–426, 2010.

[24] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. U.S.A.*, 115(34):8505–8510, 2018.

[25] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, 1989.

[26] C. Huré, H. Pham, and X. Warin. Some machine learning schemes for high-dimensional nonlinear PDEs. *Math. Comput.*, 89:1, 11 2019.

[27] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, 2014.

[28] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*, volume 23 of *Stochastic Modelling and Applied Probability*, pages 345–351. Springer Berlin, Heidelberg, 3 edition, 1999.

[29] L. Li, X. Qu, and G. Zhang. An efficient algorithm based on eigenfunction expansions for some optimal timing problems in finance. *J. Comput. Appl. Math.*, 294:225–250, 2016.

[30] Y. Maghsoodi. Mean Square Efficient Numerical Solution of Jump-Diffusion Stochastic Differential Équations. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 58(1):25–47, 1996.

[31] Y. Maghsoodi. Exact solutions and doubly efficient approximations of jump-diffusion itô equations. *Stochastic Analysis and Applications*, 16(6):1049–1072, 1998.

[32] A. M. Malyscheff and T. B. Trafalis. Natural gas storage valuation via least squares Monte Carlo and support vector regression. *Energy Syst.*, 8(4):815–855, 2017.

[33] M. Olofsson, T. Önskog, and N. L. P. Lundström. Management strategies for run-of-river hydropower plants: an optimal switching approach. *Optim. Eng.*, 2021.

[34] E. Platen. *An approximation method for a class of Itô processes with jump component*, pages 124–136. Wissenschaftliche Sitzungen zur Stochastik, WSS-01/80. Akademie der Wissenschaften der DDR, 1982.

[35] A. Porchet, N. Touzi, and X. Warin. Valuation of a power plant under production constraints and market incompleteness. *Math. Oper. Res.*, 70:47–75, 2009.

[36] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys.*, 375:1339–1364, 2018.
[37] S. Tang and X. Li. Necessary Conditions for Optimal Control of Stochastic Systems with Random Jumps. *SIAM J. Control Optim.*, 32(5):1447–1475, 1994.
[38] U. Tanielian and G. Biau. Approximating Lipschitz continuous functions with GroupSort neural networks . In A. Banerjee and K. Fukumizu, editors, *Int. Conf. Artif. Intell. Stat. AISTATS 2021*, volume 130 of *Proceedings of Machine Learning Research*, pages 442–450. PMLR, 13–15 Apr 2021.
[39] M. Thompson. Natural gas storage valuation, optimization, market and credit risk management. *J. Commod. Mark.*, 2(1):26–44, 2016.
[40] X. Warin. Nesting Monte Carlo for high-dimensional non-linear PDEs. *Monte Carlo Methods and Applications*, 24(4):225–247, 2018.
[41] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Netw.*, 94:103–114, 2017.

Department of Mathematics, University of Michigan, Ann Arbor, MI
Email address: erhan@umich.edu

Department of Mathematics, University of Michigan, Ann Arbor, MI
Email address: asafc@umich.edu

Department of Mathematics, University of Michigan, Ann Arbor, MI
Email address: nellisa@umich.edu