Diffusion-Based Forecasting for Uncertainty-Aware Model Predictive Control

Stelios Zarifis[1,2], Ioannis Kordonis[2], and Petros Maragos[1,2]

[1]Robotics Institute, Athena Research Center, Athens, Greece

[2]School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

s.zarifis@athenarc.gr, kordonis@central.ntua.gr, petros.maragos@athenarc.gr

***Abstract*—We propose *Diffusion-Informed Model Predictive Control* (*D-I MPC*), a generic framework for uncertainty-aware prediction and decision-making in partially observable stochastic systems by integrating diffusion-based time series forecasting models in Model Predictive Control algorithms. In our approach, a diffusion-based time series forecasting model is used to probabilistically estimate the evolution of the system's stochastic components. These forecasts are then incorporated into MPC algorithms to estimate future trajectories and optimize action selection under the uncertainty of the future. We evaluate the framework on the task of energy arbitrage, where a Battery Energy Storage System participates in the day-ahead electricity market of the New York state. Experimental results indicate that our model-based approach with a diffusion-based forecaster significantly outperforms both implementations with classical forecasting methods and model-free reinforcement learning baselines.**

***Index Terms*—Diffusion models, time series forecasting, model predictive control, uncertainty quantification, reinforcement learning, energy markets.**

I. Introduction

Real-world decision-making problems (for instance in energy markets, financial systems, inventory management) are characterized by high uncertainty and partial observability [1]. In such environments, the true state that actually influences the evolution of the system is unobservable. Hence, decision-makers rely on partial observations of the system. In this study, we highlight the potential of planning the control actions using a state-of-the-art probabilistic forecaster, over using traditional models or learning to act model-free, in a class of environments.

We propose a framework that uses a diffusion-based time series forecasting method to capture complex patterns in the stochastic dynamics, through probabilistic predictions. Integrating these probabilistic forecasts into a Model Predictive Control scheme, our algorithm aims to solve decision problems particularly in scenarios where data is sparse and there is no prior knowledge regarding how actions are taken. As a case study, we apply our method, to control a Battery Energy Storage System (BESS) in the day-ahead electricity market of the New York state, where price volatility creates significant profit opportunities. We found that our method outperformed all other methods we tested. This work offers a flexible framework for uncertainty-aware control using diffusion-based predictive models, applicable to various problems.

II. Background and Related Work

A. Diffusion Models

1) Denoising Diffusion Probabilistic Models: Denoising Diffusion Probabilistic Models (DDPMs) [2] are generative models that approximate a data distribution by iteratively denoising samples from a Gaussian prior. DDPMs are based on a forward process that gradually adds noise to the data and a learned reverse process that removes this noise step by step, aiming to reconstruct the initial data samples. These models became popular by achieving state-of-the-art results in modeling complex multimodal distributions in image generation tasks, thanks to their inherent high complexity and flexibility.

The fundamental idea of DDPMs is to define a diffusion process that maps a data distribution to a simple prior distribution, usually a Gaussian, through a sequence of transformations. Given an initial data sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, the forward diffusion process iteratively adds Gaussian noise to acquire increasingly noisier versions of the sample:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{1}$$

where α_t is a variance schedule controlling the noise level at each step t. This process transforms the data into a nearly Gaussian distribution as $t \to T$. To enable reconstruction, the forward process must be approximately invertible, which requires sufficiently small time steps so that the distribution shift between consecutive states remains smooth.

A reverse process aims to denoise $\mathbf{x}_T$ back to $\mathbf{x}_0$ through a learned model p_θ:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(t)). \tag{2}$$

Formulating DDPMs as latent variable generative models, their goal is to approximate the true data distribution $q(\mathbf{x}_0)$ through the learned process:

$$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T} \approx q(\mathbf{x}_0), \tag{3}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latent variables of the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$.

2) TimeGrad Model: TimeGrad [3] is an autoregressive probabilistic forecasting model that combines DDPMs with a Recurrent Neural Network to model time series distributions. TimeGrad can capture multimodal distributions and long-range dependencies in multivariate time series better than traditional forecasters. A multivariate time series is defined as $x_{i,k} \in \mathbb{R}$

for $i \in \{1, \ldots, D\}$, where k is the time index and i is the time series index, and the multivariate vector at time k is denoted as $\mathbf{x}_k \in \mathbb{R}^D$. The forecasting problem can be defined as predicting a window of future values of a multivariate time series $\mathbf{x}_{k_0:N}$ based on the past values $\mathbf{x}_{1:k_0-1}$ and possibly additional information $\mathbf{c}_{1:N}$ to encode events that affect the dynamics. In our framework, we will not use covariates, though they can be included if desired.

TimeGrad models the conditional distribution of future values given past observations:

$$q\left(\mathbf{x}_{k_0:N} | \mathbf{x}_{1:k_0-1}\right) = \prod_{k=k_0}^{N} q\left(\mathbf{x}_k | \mathbf{x}_{1:k-1}\right), \tag{4}$$

where $\mathbf{x}_k$ is the multivariate vector of the time series at time k. Each factor is approximated using a DDPM conditioned on historical data, encoded using an RNN:

$$\mathbf{h}_k = \mathrm{RNN}_\theta\left(\mathbf{x}_k, \mathbf{h}_{k-1}\right), \tag{5}$$

where $\mathbf{h}_{k-1}$ is the hidden RNN state at time $k-1$. The model learns to approximate the future time series distribution:

$$p_\theta\left(\mathbf{x}_k | \mathbf{h}_{k-1}\right) \approx q\left(\mathbf{x}_k | \mathbf{x}_{1:k-1}\right). \tag{6}$$

Here, θ includes both RNN and diffusion model parameters, which are optimized jointly. Therefore, TimeGrad's goal is to approximate the conditional distribution of the future values using the individual learned distributions of Equation (6):

$$p_\theta\left(\mathbf{x}_{k_0:N} | \mathbf{x}_{1:k_0-1}\right) = \prod_{k=k_0}^{N} p_\theta\left(\mathbf{x}_k | \mathbf{h}_{k-1}\right) \approx q\left(\mathbf{x}_{k_0:N} | \mathbf{x}_{1:k_0-1}\right). \tag{7}$$

Observing that TimeGrad's goal in Equation (7) is the exact same as DDPMs' goal in Equation (3), the authors utilized the diffusion processes to learn the target distribution.

B. Reinforcement Learning

Reinforcement Learning (RL) is a framework for learning control policies from interactions in the environment.

Model-free RL value-based methods such as Q-learning [4] and Deep Q-Networks (DQN) [5] and policy-based methods like Proximal Policy Optimization [6] have demonstrated success in complex tasks. *Model-based RL* methods use an approximate model of the environment to plan actions. These methods are motivated by scenarios with limited data, where model-free approaches often struggle to learn a robust policy due to insufficient interactions with the environment [7].

Several ideas have been recently proposed on related tasks, especially in robotics. Approaches such as [8], [9] use probabilistic models to improve robustness and efficiency of planning. Moreover, recent works such as [10]–[14] utilize diffusion models to generate state-action sequences that optimize a given goal and [15] estimates the states based on noisy observations, leading to more robust control. However, most of the proposed models are trained on a dataset of state-action pairs, which is not always available. Our method is based on a model of the system, generated by a diffusion-based forecaster, to directly optimize the action sequence, without any prior knowledge on how actions are chosen, which is suitable for environments with sparse data or where it is not possible to make multiple simulations to generate training data.

III. Methodology

A. Problem Definition

We present a method for a class of partially observable systems, where the dynamics can be separated into a deterministic and a stochastic component, and the actions of an agent do not affect the stochastic dynamics. This class of problems is interesting as it is widespread in many real-world applications, such as in energy markets, finance, inventory management, and robotics, where the systems exhibit a mix of predictable and uncertain dynamics [16]. Our method models the system as a Partially Observable Markov Decision Process (POMDP). The POMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, T, R, \Omega, O, \gamma)$, where

- $\mathcal{S}$ is the set of all possible (hidden) states,
- $\mathcal{A}$ is the set of actions available to the decision-maker,
- $T : S \times A \to \Delta(S)$ is the state transition function, with $T(s' \mid s, a)$ giving the probability of transitioning to state s' when action a is taken in state s, where $\Delta(\mathcal{S})$ denotes the probability distribution over the set of states S.
- $R : O \times A \to \mathbb{R}$ is the reward function, yielding the immediate reward $R(o, a)$ when taking action a in the observation state o.
- Ω is the set of observations that the agent can receive.
- $O : S \times A \to \Delta(\Omega)$, the observation function; $O(o \mid s', a)$, is the probability of observing $o \in \Omega$ given that the environment transitions to state s', with action a.
- $\gamma \in [0, 1)$ is the discount factor.

At each time step k, the environment is in some state $s_k \in \mathcal{S}$ (which is not directly observed by the agent). The agent selects an action $a_k \in \mathcal{A}$; as a result, the environment transitions to a new state s_{k+1} according to the probability $T(s_{k+1} \mid s_k, a_k)$, and the agent receives an observation $o_{k+1} \in \Omega$, where $o_{k+1} \sim O(o_{k+1} \mid s_{k+1}, a_k)$ and an immediate reward $r_k = R(o_k, a_k)$.

The objective of the agent at time k_0, given the history of observations, is to find a policy $\pi(o_1, o_2, \cdots, o_{k_0})$ that maximizes its expected cumulative discounted reward:

$$\underset{\pi}{\text{maximize}} \;\; \mathbb{E}\left[\sum_{k=k_0}^{\infty} \gamma^k r_k\right]. \tag{8}$$

In many practical applications (such as energy bidding, battery storage control, or inventory management) the state and dynamics are unmeasurable or unknown. For instance, in energy bidding, where participants submit bids to buy or sell electricity before the publication of the actual energy prices, it is impossible to obtain a true state, as the formation of future prices depends on factors like the decisions of the bidders, the energy demand, the weather, which are not fully predictable.

The following analysis considers problems for which we assume that the true state $s_k \in \mathcal{S}$ is not directly observed. Instead, for each time step k, the agent receives an observation $o_k \in \Omega$. Furthermore, the observations are partially deterministic and partially stochastic: $o_k = \left(o_k^d, o_k^s\right)$, where

the actions affect only the deterministic part $o^d_{k+1} = f(o^d_k, a_k)$ (for example, in the application presented in Section IV, the state-of-charge of a battery evolves deterministically while energy prices have stochastic dynamics). The agent forecasts the stochastic part of the future observations $\hat{o}^s_{k+1}, \hat{o}^s_{k+2}, \ldots$ to plan its actions.

B. Probabilistic Forecasting with Diffusion Models

To account for the inherent uncertainty and complexity that most real-world systems have, we employ TimeGrad to provide a stochastic model for the observations dynamics, by approximating the conditional probability distribution of future values given historical data:

$$q(\mathbf{o}^s_{k_0:N} \mid \mathbf{o}^s_{1:k_0-1}) \approx p_\theta\left(\mathbf{o}^s_{k_0:N} \mid \mathbf{h}_{1:k_0-1}\right). \tag{9}$$

C. Model Predictive Control with Diffusion Models

We embed the forecasting diffusion model into a Model Predictive Control (MPC) framework to determine the optimal decision sequence in the POMDP, considering the forecasts as the stochastic model of the system. MPC is a control strategy that optimizes the evolution of dynamic systems by predicting future behavior and determining the optimal control actions over a finite time horizon. . In the deterministic MPC, TimeGrad is used as a point estimator by aggregating its probabilistic forecasts. Specifically, at each decision epoch, we define the forecast operator $\mathcal{F}$ as:

$$\hat{o}^s_{k+1} = \mathcal{F}(o^s_0, \ldots, o^s_k) = \text{median}\left\{\hat{o}^{s,(i)}_{k+1}\right\}_{i=1}^{M}, \tag{10}$$

where $\{\hat{o}^{s,(i)}_{k+1}\}_{i=1}^{M}$ is the set of M forecasted future observations generated by TimeGrad (the operator may alternatively use the mean). Then, the MPC optimization problem is formulated as:

$$\begin{aligned}
\underset{a_{k_0},\ldots,a_{N+k_0-1}}{\text{maximize}} \quad & \sum_{k=k_0}^{N+k_0-1} R(\hat{o}_k, a_k) \\
\text{subject to} \quad & o_0, \ldots, o_{k_0-1} \text{ observed}, \\
& \hat{o}^d_{k+1} = f(\hat{o}^d_k, a_k),\ \hat{o}_{k_0} = o_{k_0}, \\
& \hat{o}^s_{k+1} = \mathcal{F}(o^s_0, \ldots, o^s_{k_0-1}, \hat{o}^s_{k_0}, \ldots, \hat{o}^s_k), \\
& a_k \in \mathcal{A}, \quad k = k_0, \ldots, N+k_0-1.
\end{aligned} \tag{11}$$

By aggregating TimeGrad's probabilistic forecasts, our approach handles uncertainty only at the forecast level, resulting in a deterministic MPC optimization process. In contrast, stochastic MPC (SMPC) considers multiple trajectory realizations to integrate uncertainty in the optimization process too.

D. Stochastic Model Predictive Control with Diffusion Models

To handle stochasticity in the system dynamics, which impacts the decision sequence, we extend the deterministic MPC framework by considering multiple forecast trajectories throughout the optimization process. The distribution of generated future trajectories is used to model the uncertainty.

We implement Monte Carlo SMPC, where we incorporate forecast uncertainty into the optimization process by sampling M trajectories from the future values distribution generated by TimeGrad. The goal is to maximize the expectation of the total reward, which is approximated by the average total reward of M realizations of the stochastic system. For each scenario $i \in \{1, \ldots, M\}$, the entire future trajectory from time k_0 to $N + k_0 - 1$ is generated recursively:

$$\hat{o}^{s,(i)}_{k+1} \sim p_\theta\left(\hat{o}^{s,(i)}_{k+1} | \mathbf{h}^{(i)}_k\right), \quad k = k_0, \ldots, N+k_0-1, \tag{12}$$

where p_θ denotes TimeGrad's forecast distribution. The cumulative reward for each scenario is given by:

$$\mathcal{J}^{(i)} = \sum_{k=k_0}^{N+k_0-1} R\left(\hat{o}^{(i)}_k, a_k\right). \tag{13}$$

The SMPC optimization problem is then formulated as:

$$\begin{aligned}
\underset{a_{k_0},\ldots,a_{N+k_0-1}}{\text{maximize}} \quad & \frac{1}{M}\sum_{i=1}^{M} \mathcal{J}^{(i)} \\
\text{subject to} \quad & o_0, \ldots, o_{k_0-1} \text{ observed}, \\
& \hat{o}^d_{k+1} = f(\hat{o}^d_k, a_k),\ \hat{o}_{k_0} = o_{k_0} \\
& \hat{o}^{s,(i)}_{k+1} \sim p_\theta\left(\hat{o}^{s,(i)}_{k+1} | \mathbf{h}^{(i)}_k\right), \\
& a_k \in \mathcal{A}, \quad k = k_0, \ldots, N-1.
\end{aligned} \tag{14}$$

This formulation models the uncertainty in both the forecasting and the optimization process, enabling the controller to plan more robust actions, especially when the dynamics are strongly non-linear.

IV. Results

A. Experimental Setup

We evaluate our method in the context of energy arbitrage for a region in New York State. The task is to control a Battery Energy Storage System (BESS), to make long-term profitable transactions. The environment is modeled as a POMDP. The observation $o_k = (SoC_k, p_k) \in O$ consists of the battery's state-of-charge (SoC), and the electricity prices at time k. The action $a_k \in \mathcal{A}$ represents the amount of energy flowing in or out of the BESS, as a percentage of battery capacity. The transition function T has a deterministic and a stochastic part. The battery's SoC evolves deterministically, while the prices evolve stochastically. The reward is defined as the revenue generated from energy transactions minus the cost of battery degradation, according to a battery model, inspired by [17]. The action space is defined by the restrictions of the battery, i.e. the maximum energy flow, the battery's efficiency and capacity. We use hourly electricity prices from the New York Independent System Operator to test our method. The components of the POMDP are defined as:

$$\begin{cases}
s_k \in \mathcal{S}: \text{ unknown hidden state} \\
a_k \in \mathcal{A}(o_k) = \left[-\min\left(a_{max}, \frac{\text{SoC}k}{\eta}\right), \min\left(a_{max}, \frac{1-\text{SoC}_k}{\eta}\right)\right] \\
\hat{o}_{k+1} = \left(SoC_{k+1}, \hat{o}^s_{k+1}\right) = \left(SoC_{k+1}, \hat{p}^s_{k+1}\right), \\
\qquad \text{SoC}_{k+1} = \text{SoC}_k + \eta \cdot a_k,\ \hat{o}^s_{k+1} = \mathcal{F}(\hat{o}^s_0, \ldots, \hat{o}^s_k) \\
R\left(o_k, a_k\right) = \text{Revenue}(o_k, a_k) - \text{Degradation}(o_k, a_k)
\end{cases} \tag{15}$$

Optimal operation of grid-connected battery energy storage systems over their lifetime has been also examined in [18].

B. Model Comparison and Observations

Table I summarizes the performance of our MPC methods, Table II compares our methods with MPC guided by more classical predictive models, and Table III compares our method against a set of model-free RL baselines. Figures 1–2 illustrate the control strategies, where MPC and Monte Carlo SMPC (100 realizations) are used to plan over a 3-day look-ahead horizon. The optimizer re-plans its actions daily as new prices are published, applying only the first day's actions.

a) Visualization of the Method: Figures 1–2 show that the SoC rises at low prices (red dots) and decreases at high prices (green dots), while no action is indicated by gray color.

For both implementations, the algorithm adapts to daily market updates by planning over the horizon and applying only the first day's actions, since in the next day, new prices will be published, thus more accurate forecasts will be available.

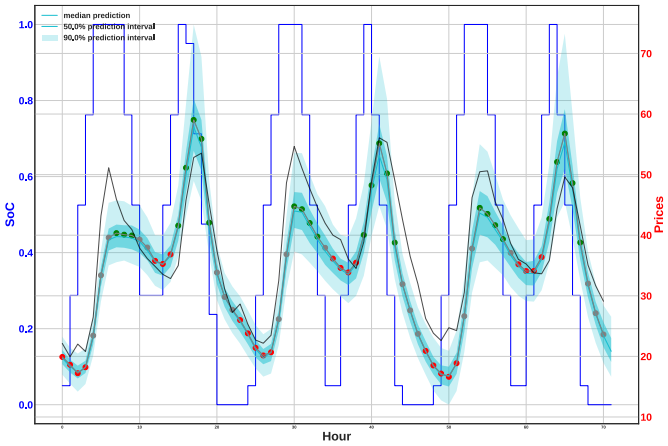

Fig. 1. *Strategy planned by the MPC Optimizer. The dotted line represents the forecasted prices, the black line represents the actual electricity prices, the blue line represents the battery's SoC, and the cyan shaded areas represent the 90% and 50% quantiles of forecast distributions.*

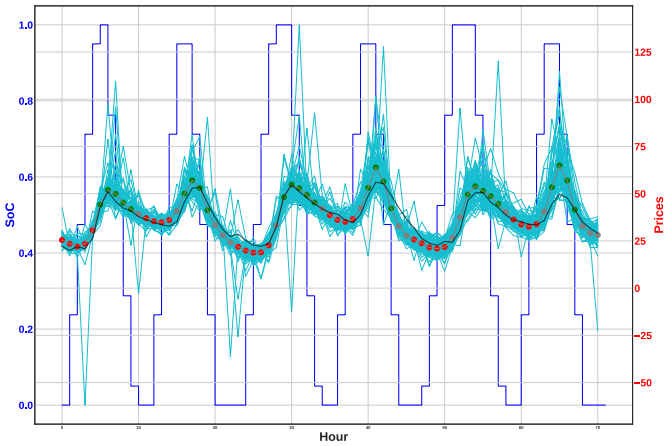

Fig. 2. *Strategy planned by the SMPC Optimizer. The cyan lines represent the generated trajectories, the black line represents the actual prices (published in the future) and the blue line represents the battery's SoC.*

b) Diffusion-Informed MPC Performance: Table I compares our implementations with two idealized ones. The Perfect MPC algorithm performs MPC on a full range of prices at once (has full knowledge of the future), while Oracle MPC uses as model the real prices (perfect forecaster) for the same horizon as the rest MPC algorithms. MPC and Monte Carlo MPC with a diffusion-based forecaster closely match the performance of the idealized benchmarks, having reward gaps of merely 3.54% and 2.22% with Oracle MPC (MPC with a perfect forecaster), respectively, confirming that diffusion models are ideal to guide the MPC optimization process.

c) Classical Model MPC vs. Diffusion-Informed MPC: When comparing forecasting models in Table II, MPC guided by a diffusion-based forecaster significantly outperforms those based on classical methods, with a 38.8% margin over the second best. Notably, the LSTM of Table II has the exact same architecture with TimeGrad's RNN, to showcase the advantage of using diffusion processes for more accurate forecasts.

It is important to note that, although the deep learning models we trained achieve better error metrics compared to autoregressive models, their ability to guide MPC algorithms is significantly worse. The reason lies in the specific application on which we test our method, since accurately predicting the peaks and valleys of energy prices is far more critical than merely achieving low error rates. The CNN and LSTM models tended to estimate the trends of the price time series sometimes with a small shift, resulting in the agent, which follows the MPC policy, being confused on when it is optimal to make transactions. This is illustrated in Figure 3, where the CNN and LSTM demonstrate better reward anticipation compared to classical models, since they provide more accurate forecasts. However, MPC based on these deep learning models does not exhibit improved performance, due to the said shifts. This underscores again the importance of accurately predicting the details in the state trajectory, hence diffusion-based forecasters are highly suitable to guide the MPC.

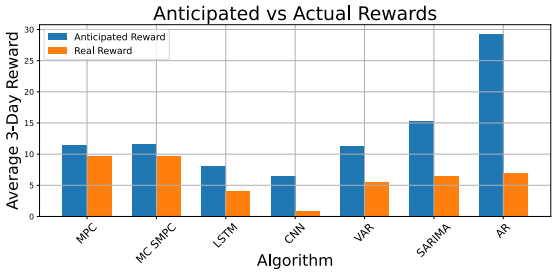

Fig. 3. Anticipated and actual rewards (3-day windows) for various models, sorted by how closely the anticipated matches the actual reward.

d) Model-Free RL vs. Model-Based Approaches: Table III shows that model-free RL implementations, even when provided with perfect knowledge of the next 12 prices as state, fall considerably behind *Diffusion-Informed MPC*. The average reward of the best DQN agent is 69.5% lower than the reward achieved by our model, highlighting the advantage of our model-based method over model-free approaches, since the system dynamics can be approximated by a model.

V. Conclusion

We defined a novel framework for uncertainty-aware prediction and decision-making in partially observable stochastic systems by integrating diffusion-based time series forecasting with Model Predictive Control. Our approach uses diffusion models to generate probabilistic forecasts of the system's stochastic dynamics, which are used to guide MPC algorithms

TABLE I
D-I MPC PERFORMANCE (HIGHER IS BETTER).

Month	Perfect MPC	Oracle MPC	MPC	MC SMPC
2018-06	99.98	97.13	90.72	92.11
2018-07	147.46	141.56	134.61	138.75
2018-08	123.11	115.31	111.62	112.43
2018-09	108.44	102.32	100.65	100.53
2018-10	104.05	95.36	103.48	102.84
2019-04	60.35	54.62	53.42	53.51
2020-12	68.85	59.31	60.58	61.61
2021-01	64.03	59.00	57.09	56.60
2021-02	109.34	108.33	95.46	96.69
2021-03	56.94	56.35	50.15	54.54
Sum	942.57	889.30	857.78	869.62
Average	94.26	88.93	85.78	86.96

TABLE II
D-I MPC VS. CLASSICAL MODELS MPC PERFORMANCE (HIGHER IS BETTER).

Month	TimeGrad MPC	AR MPC	ARIMA MPC	SARIMA MPC	VAR MPC	CNN MPC	LSTM MPC
2018-06	**90.72**	87.59	−1.28	80.25	65.84	60.47	74.84
2018-07	**134.61**	109.56	−3.11	106.86	68.81	3.59	38.13
2018-08	**111.62**	93.55	−2.55	93.4	61.02	−15.46	3.59
2018-09	**100.65**	81.44	1.33	83.38	60.77	−1.19	32.56
2018-10	**103.48**	57.71	−1.31	62.07	60.85	38.76	79.75
2019-04	**53.42**	16.45	−1.64	25.34	30.77	18.73	50.13
2020-12	**60.58**	47.05	−0.79	35.27	30.65	12.54	54.41
2021-01	**57.09**	47.51	−0.16	38.39	37.63	−2.23	42.87
2021-02	**95.46**	56.69	−4.39	36.93	56.99	−19.42	−46.61
2021-03	50.15	20.44	−0.92	22.24	28.2	−5.31	**50.88**
Sum	**857.78**	618.00	−14.82	584.14	501.54	90.48	380.54
Average	**85.78**	61.80	−1.48	58.41	50.15	9.05	38.05

TABLE III
MODEL-FREE RL AND DIFFUSION-INFORMED MPC PERFORMANCES (HIGHER IS BETTER).

	Model-Free RL (Knowledge: 12 real prices ahead)					Model-Based RL
Month	**DQN** $32 \times 64 \times 32$	**DQN** $64 \times 128 \times 64$	**DQN** $128 \times 256 \times 128$	**DQN** $64 \times 128 \times 128 \times 64$	**DQN** $64 \times 128 \times 256 \times 128 \times 64$	**TimeGrad**
2018-06	8.40	30.89	21.34	4.62	61.28	**92.11**
2018-07	14.99	53.44	30.08	8.80	45.05	**138.75**
2018-08	6.14	46.00	10.64	3.56	25.57	**112.43**
2018-09	3.87	28.21	17.99	3.07	20.14	**100.53**
2018-10	8.29	31.75	18.96	8.79	14.46	**102.84**
2019-04	3.43	18.65	10.25	2.41	6.00	**53.51**
2020-12	3.55	15.35	9.32	3.67	7.18	**61.61**
2021-01	2.91	19.37	10.47	7.01	10.62	**56.60**
2021-02	2.58	12.44	14.39	3.36	0.32	**96.69**
2021-03	1.59	13.10	7.46	6.43	5.65	**54.54**
Sum	55.76	269.15	151.80	51.02	196.27	**869.62**
Average	5.58	26.92	15.18	5.10	19.63	**86.96**

to select optimal actions. The application of our method in a real-world scenario demonstrates significant advantages over MPC with traditional forecasters and model-free RL approaches. Future work will explore further enhancements, including Scenario-Tree based MPC and a hybrid approach combining model-free and model-based methods.

REFERENCES

[1] T. Badings, T. D. Simão, M. Suilen, and N. Jansen, "Decision-making under uncertainty: beyond probabilities," *International Journal on Software Tools for Technology Transfer*, vol. 25, no. 3, pp. 375–391, Jun. 2023. [Online]. Available: https://doi.org/10.1007/s10009-023-00704-3.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," arXiv preprint arXiv:2006.11239, Jun. 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[3] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," arXiv preprint arXiv:2101.12072, Jan. 2021. [Online]. Available: https://arxiv.org/abs/2101.12072

[4] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, May 1992, doi:10.1007/BF00992698.

[5] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," arXiv preprint arXiv:1312.5602, Dec. 2013. [Online]. Available: https://arxiv.org/abs/1312.5602

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv preprint arXiv:1707.06347, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[7] L. Da, J. Turnau, T. P. Kutralingam, A. Velasquez, P. Shakarian, and H. Wei, "A Survey of Sim-to-Real Methods in RL: Progress, Prospects and Challenges with Foundation Models," arXiv preprint arXiv:2502.13187, 2025. [Online]. Available: https://arxiv.org/abs/2502.13187

[8] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A Model-Based and Data-Efficient Approach to Policy Search," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 465–472.

[9] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models," arXiv preprint arXiv:1805.12114, May 2018. [Online]. Available: https://arxiv.org/abs/1805.12114

[10] G. Zhou et al., "Diffusion Model Predictive Control," arXiv preprint arXiv:2410.05364, 2024. Available: https://arxiv.org/abs/2410.05364

[11] S. Samavi, "Safe Diffusion Model Predictive Control for Interactive Robotic Crowd Navigation," in *CoRL Workshop on Safe and Robust Robot Learning for Operation in the Real World*, 2024. [Online]. Available: https://openreview.net/forum?id=OIifSeluIk

[12] F. Djeumou, T. J. Lew, N. Ding, M. Thompson, M. Suminaka, M. Greiff, and J. Subosits, "One Model to Drift Them All: Physics-Informed Conditional Diffusion Model for Driving at the Limits," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=0gDbaEtVrd

[13] N. Botteghi, F. Califano, M. Poel, and C. Brune, "Trajectory Generation, Control, and Safety with Denoising Diffusion Probabilistic Models," arXiv preprint arXiv:2306.15512, 2023. [Online]. Available: https://arxiv.org/abs/2306.15512

[14] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-Order Sampling-Based MPC for Torque-Level Locomotion Control via Diffusion-Style Annealing," arXiv preprint arXiv:2409.15610, 2024. [Online]. Available: https://arxiv.org/abs/2409.15610

[15] Z. Yang and Y. Xu, "DMBP: Diffusion Model-Based Predictor for Robust Offline Reinforcement Learning Against State Observation Perturbations," in *Proc. 12th Int. Conf. Learn. Represent. (ICLR)*, 2024.

[16] Warren B. Powell, *Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions*, John Wiley and Sons, Hoboken, 2022, 1100 pages.

[17] D. Kumar, "Reinforcement Learning for Energy Storage Arbitrage in the Day-Ahead and Real-Time Markets with Accurate Li-Ion Battery Dynamics Model," Master's thesis, Massachusetts Institute of Technology, 2021. [Online] Available: https://hdl.handle.net/1721.1/139292

[18] I. Kordonis, A. Charalampidis, and P. Haessig, "Optimal operation of a grid-connected battery energy storage system over its lifetime," *Optimal Control Applications and Methods*, vol. 44, Nov. 2021, doi:10.1002/oca.2830.