Data-Driven Reinforcement Learning for Optimal Motor Control in Washing Machines

Chanseok Kang, Guntae Bae, Daesung Kim, Kyoungwoo Lee, Dohyeon Son,
Chul Lee, Jaeho Lee, Jinwoo Lee, Jae Woong Yun
AI Lab, LG Electronics
Seoul, Korea
{chanseok.kang, guntae.bae, daesungc.kim, kyoungwoo.lee, dohyeon.son,
clee.lee, jaeho56.lee, jinwoo.lee, jaewoong.yun}@lge.com

***Abstract*—In this paper, we address the challenge of developing advanced motor control systems for modern washing machines, which are required to operate under various conditions. Traditional system designs often rely on manual trial-and-error methods, limiting the potential for performance enhancement. To overcome this, we propose a novel continual offline reinforcement learning framework, specifically tailored to improve balance maintenance during the dehydration cycle of washing machines. Our approach introduces a delayed online update mechanism that leverages accumulated transition data from certain periods of online interaction. This method effectively circumvents the distribution shift problem commonly encountered in offline reinforcement learning. Our empirical results demonstrate a substantial improvement, with an average increase of nearly 16% in load balancing efficiency across various tasks, including those involving different types of laundry. This research not only enhances the applicability of reinforcement learning in industrial settings but also represents a significant step forward in the development of smart appliance technology.**

***Index Terms*—Washing Machine, Offline RL, Industrial AI**

I. Introduction

A washing machine is one of the most ubiquitous home appliances, globally essential for everyday life. Modern washing machines are expected to perform efficiently across various washing cycles-regular, heavy-duty, and delicate-catering to different fabric types. They must effectively clean and remove stains, consume minimal energy and water, and be gentle on fabrics. Additionally, with growing consumer demands, there is now an increasing expectation for these machines to operate noiselessly and with quicker washing times.

In response to these diverse requirements, home appliance manufacturers have been innovating in washing machine design, especially in the area of motor control systems. Traditionally, these systems are developed through extensive trial-and-error experiments using real machines, often following a 'user-in-the-loop' approach. However, this method is increasingly showing its limitations in enhancing performance, primarily due to the complex nature of washing machine dynamics and the absence of a comprehensive simulator for accurate modeling.

Recognizing these challenges, our research explores the application of deep reinforcement learning (RL) techniques, successful in other fields such as robotics [16], [20], [27], to the realm of washing machine motor control. This approach

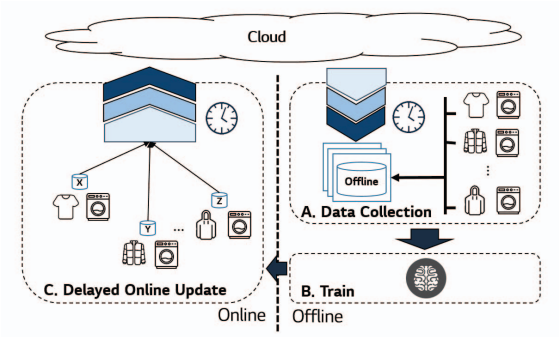

Fig. 1: **Structure of Delayed Online Update workflow.** Each step is described in IV and the overall figure shows example of update iteration, including data collection and deployment.

is a significant departure from traditional methods, offering a potential solution to the complexities that current development processes face. However, constructing real-world evaluation models essential for RL algorithm design is challenging. [7] There are no simulator that can accurately represent the intricate dynamics of washing machines, including multi-body modeling and vibration analysis, which is a critical aspect of this research.

Applying reinforcement learning to identify optimal policies for motor control is inherently time-consuming. Moreover, models trained with offline data often fail to perform adequately in online settings, a phenomenon known as the "Data Distribution Shift" problem.

In the context of household appliances, post-deployment performance enhancements are further complicated by the low-power Micro Controller Units (MCUs) that manage these devices, posing a barrier to on-device learning. By harvesting and utilizing data from appliances operating in diverse real-world conditions, there is an opportunity to improve device performance post-deployment without requiring additional computational resources.

To that end, our approach is to expand our training datasets with real-world environmental data, an effort that not only aims to bridge the divide between theoretical modeling and

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00083

practical application but also enhances the practical viability of our solutions. This strategy also holds the potential to provide the adaptive capabilities of household appliances, offering a new paradigm in smart appliance development.

In this paper, we investigate the problem of how to construct offline deep reinforcement learning models to maintain balance of laundries during dehydration. Consumers of washing machines often complain about banging sounds during the spin cycle and this is mainly due to an unbalanced load that has caused significant imbalance. When dirty clothes are put into a washing machine, these are not evenly distributed in the drum. In other unexpected instances, heavy items mixed with lighter ones can cause it to spin unevenly. Even though our proposed offline deep reinforcement learning models are motivated by other previously studied reinforcement learning problems like [14], [16], [21], [28], to our knowledge, this is the first instance of applying offline reinforcement learning not only in the washing machine domain but also in the broader field of household appliances manufacture. Our problem domain is distinctive enough from other domains in such a way that we had to design an original, yet unique reward function for RL and carefully modify and adapt state-of-the-art RL approach to our problem. More specifically, our contributions can be summarized as:

- We conduct the first formal investigation, to the best of our knowledge, into maintaining the balance of laundries during dehydration in washing machines using a Markov Decision Process (MDP) model with discrete actions. Our model considers the operational information of motor control gathered from washing machines.
- Our experiments demonstrate that our studied offline RL model effectively improves the success rate of dehydration and corresponding Time-to-Success (TTS) in real-world washing machine settings. Specifically, we observed that our delayed online update mechanism leads to performance enhancements as more data is incorporated.

The rest of the paper is organized as follows. In Section II, we formally formulate the problem of maintaining balance during dehydration in washing machines as Markov Decision Process (MDP). In Section III, we present other use-cases for applying AI technology in washing machine and previous works related to offline RL algorithms that are relevant to our problem. In Section IV, we introduce our iterative delayed online update process to solve our offline RL problem. In Section V, we present our experimental results, showing that our proposed approaches are highly effective in improving the balance maintenance of load during dehydration in washing machines, which surpasses traditional human-derived methods. And we conclude our approach in Section VI.

II. Problem Setup and Background

In this section, we formally define our problem of maintaining balance during dehydration in washing machines.

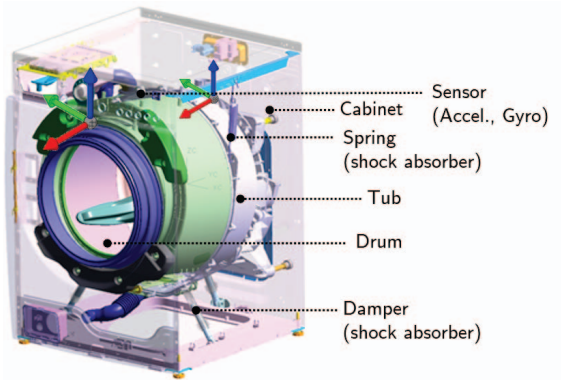

Fig. 2: **Structure of Front-Loader Washing Machine.** Most of the commercial washing machines have a similar hardware structure, consisting of drums, suspension systems and sensors.

A. Dehydration in Washing Machines

In this paper, we investigate the problem of maintaining balance of laundries during the dehydration process. General front-loader (FL) washing machines have the drum installed inside the cabinet as seen in Figure 2. The drum, whose purpose is to wash and dehydrate its laundry load, is controlled by the motor in the washing machine. The centre of mass of the laundry load will not usually lie on the axis of symmetry of the drum and thus there will be an out-of-balance load (OOBL) when the drum rotates, causing its motion to be eccentric. This eccentric motion can cause unexpected vibrations due to the friction between the drum and the cabinet, resulting into unpleasant banging noise in washing machines. One method to reduce these vibrations during the dehydration process is the installation of a suspension system in the drum. However, adding such a mechanism incurs additional costs. Another approach to mitigate vibrations is to evenly distribute the laundry within the drum during the dehydration cycle. While this method may not completely eliminate vibrations in OOBL situations, it can effectively reduce them. This approach presents a cost-effective solution to address the vibration issue, leveraging strategic laundry distribution within the drum to counteract the effects of imbalance.

B. Markov Decision Process (MDP) Framework

Most of the RL research works heavily depend on the quality of the simulator that is integrated with the given RL environment [6], [26], [37], [40]. For most of the real world cases, there is no reasonably well functioning simulator (like our case) or an approximated model to properly emulate the corresponding dynamics [7], [8]. Hence, for our problem of maintaining balance of laundries during dehydration in washing machines, we design a specific Markov Decision Process (MDP) [35] to train its optimal motor control during dehydration. We represent our MDP as a 5-element tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathcal{R})$, where $\mathcal{S}$ is state space; $\mathcal{A}$ is action space; $\mathcal{P}$ is latent dynamics, and $\gamma \in [0, 1)$ is a discount factor;

Reward function $r(s'|s,a)$ is the reward for transitioning from a state s to next state s' when take action a. The objective of our problem is to find the optimal policy π^* that maximizes the cumulative sum of the expected reward, i.e., $Q^*_\pi(s,a) = \max_\pi Q_\pi(s,a)$. $Q_\pi(s,a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k}|s,a]$.

1) State Space: We design the state space $\mathcal{S}$ of our MDP to represent the internal dynamics of a washing machine. As mentioned before, there are lots of unknown variables that may affect the dehydration process. So we assume that our environment is laid into Partial Observable Markov Decision Process and select features to represent its dynamics: (a) displacement of 3 axis on the front and rear sides, (b) gyro of 3 axis on the front and rear sides, (c) acceleration of 3 axis on the front and rear sides (d) motor state in terms of its rotational speed and force. Displacement is defined as the distance from its axis center and it is used as an amplitude of vibration. For example, if the drum is rotated with unbalanced circumstance, displacement value is large and occur huge vibration. When the drum rotates, the displacement must be lower than the minimum boundaries, let's say from 10 mm to 100 mm, in both the front and rear sides. Note that, every state in our MDP is captured from a physical sensor installed in washing machines to monitor washing status.

2) Action Space: The action space $\mathcal{A}$ of MDP is designed for a low-level motor control to change revolution per minutes (RPM). The agent can request one of two actions: UP or DOWN. When the UP action is taken, RPM is increased with a specific value, let's say 5 RPM, otherwise decreased. Similar to other real world problems [8], this RPM value is different from the actual RPM since the response time and system delays exist during the motor's function in practice. One property of event handling in washing machines that is peculiar is that it is polling in nature. That is, event interactions occur periodically, as opposed to instantly, because every washing machine action is taken with fixed time boundaries.

3) Reward Function: Our objective is to rotate the drum to the target RPM while ensuring that the displacement values stay within a predefined range. We assume that our MDP model is characterized by long-horizon and sparse rewards. If the drum reaches the target RPM within the given time while keeping its displacement within the set range, the agent receives a $+1$ reward. Conversely, if the drum fails to maintain balance during the RPM increase, leading to displacement values outside the predefined range, the episode ends with a -1 reward. Furthermore, if the drum maintains balance but fails to reach the target RPM within the allotted time, this is classified as a 'timeout', resulting in a 0 reward. In this case, there might exist a sub-optimal trajectory and hence we cannot assess its goodness. More specifically, the reward function $\mathcal{R}$ for our MDP is defined as:

$$\mathcal{R} = \begin{cases} +1, & \text{Successfully reach the target RPM within the allocated time} \\ -1, & \text{Fail to maintain the balance during RPM increase} \\ 0, & \text{Not reach the target RPM within the allocated time} \end{cases}$$

When the motor control is sub-optimally controlled, we cannot guarantee the quality of a trajectory at a given timestep. This is because it may result in unexpected outcomes. Therefore, we have designed our reward function with a sparse setting.

4) Dynamics: Based on our understanding, the type and weight of laundry loaded into a washing machine significantly influence the displacement changes and the load on the motor. Consequently, we have approached each laundry item as a distinct problem. In reality, laundry used in households typically consists of various items mixed together. Therefore, we have conducted experiments on several representative types of individual laundry items as well as on their combinations to simulate real-world scenarios.

III. Related works

AI in Washing Machine Numerous attempts have been made to leverage sensor data from washing machines for the application of artificial intelligence techniques. One notable example involves using deep learning to estimate the load weights and fabric softness, subsequently selecting optimal wash motions for different fabric types [10]. Additionally, deep learning has been utilized to detect abnormalities in washing machine operations [5], [33]. The fundamental motivation for applying AI to washing machines stems from the vast array of environments in which they operate, making it impractical to manually address every possible scenario. Our research builds upon existing studies [4] that apply reinforcement learning to washing machines, especially on the dehydration step. We aim to enhance load balance efficiency by utilizing offline datasets collected in various settings.

Offline RL. Offline RL [15], [24] is an off-policy RL approach that uses offline data. The effectiveness of offline RL has been shown in various application domains such as robot manipulation [20], [27], [31], natural language processing [18], [19], and healthcare [32], [38]. Offline RL usually suffers from distribution shift or extrapolation error as the learned RL model takes out-of-distribution actions during testing, especially when it encounters unseen offline data. As a consequence, there approaches have been proposed to deal with extrapolation errors with policy constraints [3], [15], [23], [41] and uncertainty control [1], [34]. In this paper, we tries to find the proof-of-concept of offline RL approach in real world application, so we choose naive batch RL approach for the offline baseline. Our approach is similar with works in [3] that evaluation process is happened just once with online setting. This process is probably abstract of policy iteration [35] in offline setting, or "Growing Batch Learning" concept from [24].

One emerging topic in offline RL is Offline-to-Online Learning [2], [39], [42], which incorporates online fine-tuning to address these challenges. However, such fine-tuning requires significant computational resources and online interaction, making it impractical for deployment in real-world settings.

IV. Framework overview and procedure

The general workflow is illustrated in Figure 1. The development and deployment process of our model takes place primarily in two locations: the offline and the online. Initially, we gather a dataset at the offline place such as the manufacturing site and use it to train the offline RL model. This model is then deployed into washing machines, usually installed in households. In this (online) setting, the model gathers additional data, which likely includes previously unseen states such as various types of laundry and operational information like installation status. This new dataset is then utilized for further refining and re-training the model.

The procedure for learning iteration has the following steps which are described in following section:

A. Data Collection
B. Train model with Offline RL
C. Delayed Online Update

A. Data Collection

Constructing high-quality training data is essential for the success of an offline RL algorithm, particularly when it comes to capturing the diversity of environments for effective generalization. Unlike the multitude of washing machines installed in households, the number available at manufacturing sites for data collection is limited, resulting in a constrained amount of training data. The data can originate from various sources, including manual experiments conducted by humans and certification tests. To highlight the diversity and quality of our dataset, we employed an online RL strategy with a specific exploration approach for data collection, ensuring a rich and varied dataset.

B. Train model with Offline RL

In our framework, we primarily utilize the state-of-the-art Rainbow DQN algorithm [17], adapted for an offline setting. Rainbow DQN integrates several DQN extensions: Double Q-Learning, Prioritized Experience Replay (PER), Dueling Network, Multi-step Learning, Distributional RL, and Noisy Net. However, in our adapted model, PER and Noisy Net extensions are omitted due to their incompatibility with our settings. While the agent cannot explore and collect new experience in offline RL, it has the risk of overfitting to the fixed dataset. PER might exacerbate this if the prioritization mechanism overly focuses on a subset of experiences, leading to a lack of generalization. And Noisy Net might increase the complexity of environment. The specific design choices of our network are detailed in Table I.

Hyperparameter	Value
Layers	[1024, 512, 256, 256]
Double Q-learning	Enabled
PER	Disabled
Dueling Network	Enabled
N-step	3
Distributional RL	C51
Noisy Net	Disabled

TABLE I: Hyperparameters used in the Rainbow DQN.

Policy selection in offline RL is crucial, involving the careful choice of hypothesis classes and hyperparameters for function approximators to learn policies. This selection is vital to prevent overfitting and ensure overall performance quality [22], [29]. Traditional off-policy evaluation methods, such as Fitted-Q Evaluation [25], Doubly Robust Policy Evaluation [36], and Importance-Sampling based Method [30], are not effective in our offline RL setting as they fail to accurately represent the real-world behavior of the RL algorithm. To address the challenge of policy selection, we employ a simple heuristic: selecting an epoch value that demonstrates the highest average success rate. Mathematically, this is represented as:

$$\underset{epoch \in \{E_1, \ldots\}}{\arg\max} \frac{1}{N} \sum_{t=1}^{N} success\_rate_t(epoch)$$

where $success\_rate$ indicates the success rate for each laundry load, N is the total number of laundry loads, and E is candidate epoch for the policy selection.

C. Delayed Online Update

Mentioned in III, achieving generalization across diverse scenarios in offline RL is a formidable challenge. This difficulty stems from the need for high-quality data that can accurately represent a wide range of environments. So far, recollecting such data post-deployment has been nearly impossible due to the connectivity.

Algorithm 1 Delayed Online Update

▷ Initial Data Collection
$\mathcal{D}_{\text{off}} \rightarrow \emptyset$
for task i = 1, 2, ... **do**
    Collect $\mathcal{D}^i$ in various ways
     (e.g. Online RL with exploration)
    $\mathcal{D}_{\text{off}} \leftarrow \mathcal{D}_{\text{off}} \cup \mathcal{D}^i$
end for
▷ Delayed Online Update
for phase = 1, 2, ... **do**
    Train the π from $\mathcal{D}_{\text{off}}$ with Rainbow DQN
    Deploy π in real environment with $\mathcal{D}_{\text{edge}} = \emptyset$
    for task i = 1, 2, ... **do**
        Evaluate π with online manner
        Store $(\mathbf{S}, \mathbf{A}, \mathbf{S}', \mathbf{R})$ into $\mathcal{D}_{\text{edge}}$
    end for
    Wait until $\mathcal{D}_{\text{edge}}$ has specific data amount
    $\mathcal{D}_{\text{off}} \leftarrow \mathcal{D}_{\text{off}} \cup \mathcal{D}_{\text{edge}}$
end for

Recently, several manufacturers have begun to introduce Internet-of-Things (IoT) frameworks [9], [13] that facilitate

communication between users and devices. This development opens up the possibility of continuously enhancing model performance by iteratively retraining with data collected from varied environments and redeploying the updated models. This approach is particularly promising for household appliances, which are ubiquitous in homes and thus can generate a vast amount of data for performance improvement.

A key element in this process is the Delayed Online Update (DOU) mechanism. This mechanism, described in Algorithm 1, refers to the overall process of intermittently transmitting collected data through the framework and updating the model, even in the absence of constant online interactions. In this context, the term 'delay' refers to the period during which the washing machine collects data in the real environment until a sufficient amount has accumulated to justify transmission. This strategy is particularly advantageous in the context of household appliances, which typically have limited computing power. As long as the data transmission and model redeployment processes are well-defined, there is substantial potential for application, even with the computational limitations of these devices.

V. Experiment

We conducted experimental evaluations of our RL framework on actual washing machines. For these experiments, we utilized standard commercial washing machines, modifying their control boards to enable motor control via our RL models and to collect data on drum status. These modifications were necessary as the machines lacked inherent network functionality for model interaction.

(a) Jean (b) T-Shirt (c) Towel (d) Jeans Towel (e) Towels Jumper A (f) Towels Jumper B

Fig. 3: **Laundry types.** 6 representative laundry types are considered as the set of problems. Each laundry has unique fabric, shape, and weight.

Our tests focused on six representative types of laundry loads, as depicted in Figure 3: a) Jean, b) T-shirt, c) Towel, d) 3 Jeans + Towel, e) 3 Towels + Jumper A, and f) 3 Towels + Jumper B. In each experiment, we loaded one of these laundry types into a machine, gradually increased the drum's rotation speed from zero to its maximum, and then controlled the rotation level for dehydration using our RL algorithm. We considered the RL algorithm successful if it completed an episode with a reward of $+1$, and -1 otherwise.

The performance of our model was assessed based on the overall success rate for each washing machine, defined as the ratio of successful trials to the total number of trials. To ensure consistency and minimize bias, each washing machine was configured identically. Additionally, as the water absorption in the laundry could influence dehydration, we conducted a rinse process in each episode to standardize water content.

A. Efficacy of delayed online update

In our study, we evaluated the performance of our RL framework after each update iteration against a baseline model trained with data collected from a manufacturing site. To efficiently assess the selected model in an actual washing machine, we shortlisted six candidate models through policy selection. Following the evaluation of these candidates, the data from the model with the highest success rate was merged into the existing offline dataset for retraining from scratch. This process was repeated twice in our experiment, with the results illustrated in Figure 4.

The data presented in Figure 4 clearly demonstrates that the average success rate of the models, post-application of the Delayed Online Update (DOU) mechanism, consistently surpasses that of their predecessors. Specifically, after two iterations of updates, the model exhibited an average success rate increase of **16%** compared to the initial offline RL model. Notably, the success rate for T-shirts improved by **21%** after two rounds of the DOU process. These findings underscore the effectiveness of the delayed online update approach, as it significantly enhances the performance of the basic offline RL framework. This outcome aligns with our expectations, as the iterative use of high-quality evaluation data broadens the exploration capabilities of the RL algorithms. While this method may not capture every possible dynamic of washing machine operation, it effectively extends the model's generalizability and performance.

B. Model Performance Against Human-Defined Standards

We further investigated the performance differences using the average Time-to-Success (TTS) metric – the average duration it takes to escalate from the starting RPM to its maximum. TTS is the formal metric of dehydration in manufacture process. Typically, lower TTS correlates with a higher success rate.

For this, we deployed our final model, identified as the best candidate from DOU_2, and compared its performance with a human-defined rule. It is important to note that the human-defined rule (**Human** in Table II), developed through extensive trial-and-error experiments by domain experts, is widely used in mass production. The comparative experimental results are presented in Table II.

Although the TTS for some laundry types was slower compared to the human-defined rule, these results clearly show that the best candidate model from DOU_2 generally outperforms the human-defined rule by **15.2%**. Notably, the laundry type '3 Towels + Jumper B' showed a significantly larger improvement margin compared to others. This specific load had previously posed considerable challenges for human experts, who attempted to find an ideal control through repetitive experimentation. However, through the application of the DOU process, we managed to enhance performance significantly. This result demonstrates that even tasks which

are difficult for humans to solve can be effectively addressed through our process, showcasing the potential of machine learning to improve performance where human efforts fall short.

Laundries	Human	DOU_2 (best)	Δ
Jean	1	0.71	
T-shirt	0.3	0.47	
Hoodie	1.43	1.98	
3 Jeans + Towel	1.92	1.38	
3 Towels + Jumper A	2.39	2.45	
3 Towels + Jumper B	3.13	1.62	
Average	1.70	1.43	**-15.2%**

TABLE II: Normalized TTS metric comparison. The TTS for Jean is used as the baseline for normalization, with other metrics divided accordingly. Hoodie is included for the formal qualification control test. Lower scores indicate better performance.

C. Discussion of Limitations

At the heart of our approach is the reliance on extensive and diverse data sets. The efficacy of the reinforcement learning model is contingent on the quality and variety of the data it is trained on. In real-world scenarios, this poses a challenge. Washing machines encounter a myriad of fabric types, load sizes, and user preferences. The current model, while robust, may not fully encapsulate the vast spectrum of real-life scenarios, potentially limiting its adaptability and effectiveness.

Additionally, most offline RL models are evaluated in simulators, which takes considerably less time, but finding an optimal model for real-world environments like washing machines involves a significant time investment. This is why we performed the Design of Experiments only twice, as the time-intensive nature of this process limits more frequent iterations.

Long-term stability and maintenance of reinforcement learning systems in dynamic and evolving real-world environments also remain an area of concern. As usage patterns change and machines age, the RL model would require continuous updates and retraining to maintain optimal performance, posing a challenge in terms of maintenance and user experience.

VI. Conclusion

In this paper, we address the challenge of maintaining load balance during the dehydration cycle in washing machines through reinforcement learning (RL) algorithms. Our experimental results demonstrate the effectiveness of our proposed RL framework, which incorporates a Delayed Online Update mechanism. This approach not only shows notable improvements in success rate and Time-to-Success during the dehydration process but also exhibits strong generalization capabilities. Moreover, this framework, a practical alternative to the computationally intensive Offline-to-Online learning solutions recently proposed for the Offline RL problem, is especially significant for augmenting the performance of low-resource devices. We have successfully followed the same steps in the production level and improved dehydration performance to a real, mass-produced washing machine [11].

To our knowledge, this is the pioneering instance of applying Offline RL to a household appliance on a global scale. Moreover, this framework opens up the potential for progressively enhancing the performance of washing machines, leveraging their connectivity features as outlined in [12].

Acknowledgment

This research was supported by LG Electronics Home Appliance & Air Solution Division. We thank Jinho Kim, Dongsoo Kang and Jinyoung Park from Washing Machine Advanced R&D Team who provided insight and expertise in Washing Machine.

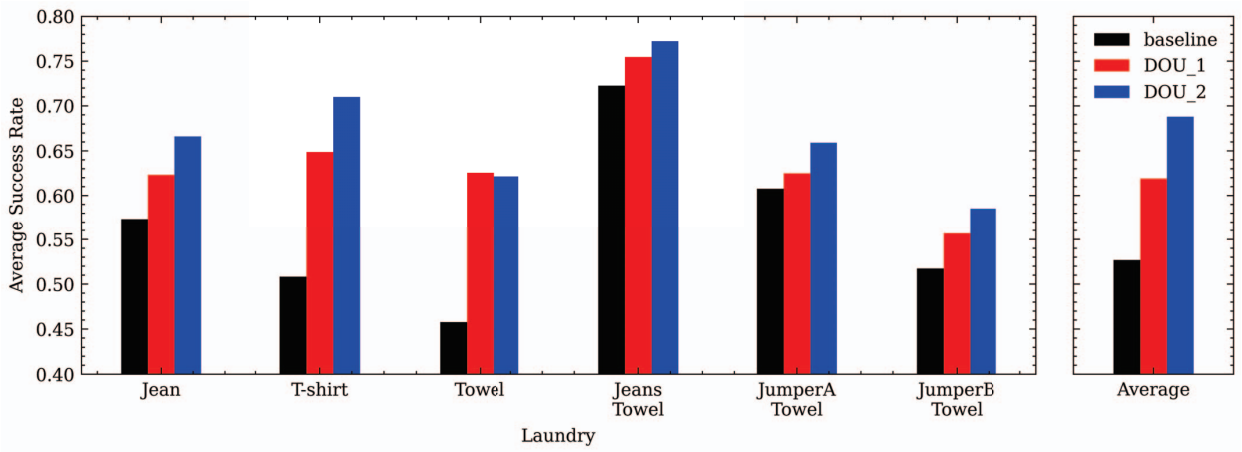

Fig. 4: **Average Success Rates Across Update Iterations.** Illustrates the average success rates for the initial offline RL model (baseline) and after one and two update iterations (DOU_1, DOU_2). The bar on the far right summarizes the average success rates across all six laundry types.

REFERENCES

[1] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

[2] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. *arXiv preprint arXiv:2302.02948*, 2023.

[3] David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. Offline RL without off-policy evaluation. *Advances in Neural Information Processing Systems*, 34:4933–4946, 2021.

[4] Dongsoo Choung and Guntae Bae. Intelligent washing machine and method for controlling ball balancer using the same, 2019. US11359319B2.

[5] Dongsoo Choung and Guntae Bae. Method and apparatus for inspecting defects in washer based on deep learning, 2020. US20210042618A1.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[7] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. An empirical investigation of the challenges of real-world reinforcement learning. *arXiv preprint arXiv:2003.11881*, 2020.

[8] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[9] LG Electronics. LG ThinQ. https://www.lg.com/us/lg-thinq.

[10] LG Electronics. LG AI direct drive washing machine: Smart convenience redefined. https://www.lg.com/in/magazine/lg-ai-direct-drive-washing-machine-smart-convenience-redefined, 2023.

[11] LG Electronics. LG offers one-stop laundry solution with new second-gen LG signature washer-dryer at IFA 2023. https://www.lgcorp.com/media/release/26653, 2023.

[12] LG Electronics. LG ThinQ UP 2.0 shifts paradigm for home appliances to personalization and servitization. https://www.lgcorp.com/media/release/26656, 2023.

[13] Samsung Electronics. SmartThings. https://www.smartthings.com/.

[14] Mehdi Fatemi, Taylor W Killian, Jayakumar Subramanian, and Marzyeh Ghassemi. Medical dead-ends and learning to identify high-risk states and treatments. *Advances in Neural Information Processing Systems*, 34:4856–4870, 2021.

[15] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[16] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018.

[17] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[18] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

[19] Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Shane Gu, and Rosalind Picard. Human-centric dialog training via offline reinforcement learning. *arXiv preprint arXiv:2010.05848*, 2020.

[20] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

[21] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[22] Ksenia Konyushova, Yutian Chen, Thomas Paine, Caglar Gulcehre, Cosmin Paduraru, Daniel J Mankowitz, Misha Denil, and Nando de Freitas. Active offline policy selection. *Advances in Neural Information Processing Systems*, 34:24631–24644, 2021.

[23] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Adaptation, Learning, and Optimization*, pages 45–73. Springer Berlin Heidelberg, 2012.

[25] Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712. PMLR, 2019.

[26] Oleh Lukianykhin and Tetiana Bogodorova. Modelicagym: applying reinforcement learning to modelica models. In *Proceedings of the 9th International Workshop on Equation-based Object-oriented Modeling Languages and Tools*, pages 27–36, 2019.

[27] Ajay Mandlekar, Fabio Ramos, Byron Boots, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Dieter Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4420. IEEE, 2020.

[28] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[29] Tom Le Paine, Cosmin Paduraru, Andrea Michi, Caglar Gulcehre, Konrad Zolna, Alexander Novikov, Ziyu Wang, and Nando de Freitas. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

[30] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[31] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, pages 1154–1168. PMLR, 2021.

[32] Susan M. Shortreed, Eric Laber, Daniel J. Lizotte, T. Scott, Stroup Joelle, Pineau Susan, A. Murphy, S. M. Shortreed, J. Pineau, J. Pineau, E. Laber, D. J. Lizotte, S. A. Murphy, E. Laber, D. J. Lizotte, and S. A. Murphy. Informing sequential clinical decision-making through reinforcement learning: an empirical study. In *Machine Learning*, 2011.

[33] Yusun Shul, Wonjun Yi, Jihoon Choi, Dong-Soo Kang, and Jung-Woo Choi. Noise-based self-supervised anomaly detection in washing machines using a deep neural network with operational information. *Mechanical Systems and Signal Processing*, 189:110102, 2023.

[34] Aaron Sonabend, Junwei Lu, Leo Anthony Celi, Tianxi Cai, and Peter Szolovits. Expert-supervised reinforcement learning for offline policy learning and evaluation. *Advances in Neural Information Processing Systems*, 33:18967–18977, 2020.

[35] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[36] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

[37] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033. IEEE, 2012.

[38] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2447–2456, 2018.

[39] Shenzhi Wang, Qisen Yang, Jiawei Gao, Matthieu Gaetan Lin, Hao Chen, Liwei Wu, Ning Jia, Shiji Song, and Gao Huang. Train once, get a family: State-adaptive balances for offline-to-online reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[40] Cathy Wu, Abdul Rahman Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, 2021.

[41] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[42] Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.