Aneesh Muppidi
aneeshmuppidi@college.harvard.edu

Harvard John A. Paulson School of Engineering and Applied Sciences

Vision Based Pendulum Control using Diffusion Policy

Aneesh Muppidi
ENG-SCI 158 Final Project Report
Harvard College
Cambridge, MA, USA

I. Introduction and Motivation

The challenge of stabilizing a pendulum in an upright position, commonly known as the inverted pendulum problem, is a quintessential benchmark that encapsulates the intricacies of physical dynamics and control theory. This problem becomes significantly more complex when the controller is expected to operate solely based on visual inputs, a task that demands an intricate understanding of the underlying physical system and robust control strategies. Vision-based pendulum control is thus not only a testbed for control algorithms but also a stepping stone towards more advanced applications in robotics and autonomous systems where visual feedback is paramount.

Recent developments in machine learning have introduced the concept of Diffusion Policy (1; 2; 3), a novel approach that has shown remarkable effectiveness in vision-based control tasks involving multimodal and high-dimensional action spaces, particularly in the domain of robotic control. The essence of Diffusion Policy lies in its ability to learn and iteratively refine actions through a process akin to stochastic denoising (4). This method, as detailed in the work "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion" by Chi et al., (1) represents a significant leap forward in the application of generative models to control systems.

In personal communications with Cheng Chi, the lead author of Diffusion Policy, it became apparent that the pendulum control problem could serve as an almost adversarial challenge to this novel control strategy. The pendulum system is inherently a second-order dynamic system with fast dynamics, where the control algorithm must account for position, velocity, and acceleration from visual sequences alone. Additionally, the one-dimensional nature of the action space stands in stark contrast to the high-dimensional spaces where Diffusion Policy has traditionally excelled.

Thus, my project sought to explore the application of Diffusion Policy to this classic control problem. I hypothesized that the advantages of the diffusion process, such as handling multimodal action distributions and temporal complexity, could be leveraged in the dynamically rich environment of pendulum control, even with its simplicity and low-dimensional action space.

Through this exploration, I sought to ascertain whether the sophisticated mechanisms of Diffusion Policy could be simplified and scaled down to address the nuances of the pendulum problem, potentially paving the way for broader applications in vision-based control strategies across both simple and complex dynamic systems. A visual summary of the model architecture can be seen in **Figure 1**.

However, as the project concluded, it became clear that the Diffusion Policy was unable to adapt effectively to this adversarial example of a second-order dynamic, one-dimensional action space problem. The key insight from the investigation was that the problem required long-term stability rather than merely achieving a state-specific goal for success. This finding underscored the challenges in applying diffusion-based approaches to control tasks demanding sustained stability, marking a significant deviation from the initial expectations set by the prior applications in more goal-oriented scenarios.

II. Problem Setup and Data Collection (DQN)

A. The Pendulum Control Problem

The exploration of the pendulum control problem is conducted within OpenAI Gym's `Pendulum-v1` environment (5), a simulated setting recognized for benchmarking reinforcement learning algorithms and control systems. The environment encapsulates an inverted pendulum swing-up task, a fundamental problem in control theory where the objective is to exert torques to swing the pendulum into an upright position and stabilize it against gravity.

B. Environment Dynamics

The pendulum, modeled as a rigid body, is attached to a fixed pivot point, with the other end free to swing in a two-dimensional plane. The state of the pendulum is characterized by its angle and angular velocity, with the angle being measured from the vertical upright position. The system is initialized in a random state, with the pendulum at a random angle and angular velocity. The dynamics are governed by the following elements:

- **Action Space**: The action is a one-dimensional continuous space, represented by a `ndarray` with shape $(1,)$, indicating the torque applied to the pendulum's free end. The torque range is limited to a minimum of -2.0 Nm and a maximum of 2.0 Nm, allowing the controller to apply forces in either clockwise or counter-clockwise directions.

Aneesh Muppidi
aneeshmuppidi@college.harvard.edu

Harvard John A. Paulson School of Engineering and Applied Sciences

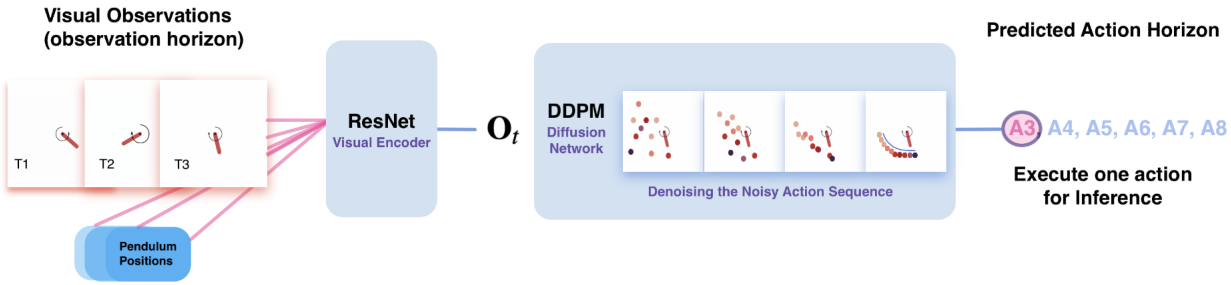

Fig. 1. The Vision based Pendulum Control Diffusion Policy. A series of visual observations T_1, T_2, T_3, ... along with their positional states are projected to a latent space and concatenated as O_t. This observation is used by the DDPM noise network to denoise an action prediction horizon of the true action sequence taken after observing the observation.

- **Observation Space**: The state of the pendulum is represented by a three-dimensional continuous space, consisting of the x and y coordinates of the pendulum's free end (derived from the cosine and sine of the angle) and the angular velocity. The x and y coordinates are constrained between -1.0 and 1.0, while the angular velocity ranges from -8.0 to 8.0 rad/s.
- **Rewards**: The reward function is designed to quantify the proximity of the pendulum to the desired upright position and is inversely proportional to the square of the angle from the vertical, the square of the angular velocity, and the square of the applied torque. The reward structure incentivizes minimal angle deviation, angular velocity, and torque application. **Note the reward is only used to train the DQN but not used to train the diffusion policy itself.**
- **Starting State**: Each episode begins with the pendulum in a randomly chosen angle ranging from $-\pi$ to π and a random angular velocity within $[-1, 1]$.

C. Data Collection for Training

To generate a comprehensive dataset for training the Diffusion Policy and vision encoder, I employ a pre-trained DQN controller (6) to interact with the environment. The DQN controller is adept at complex trajectory planning, such as swinging the pendulum to alternating sides to accumulate sufficient energy for the swing-up maneuver. This strategy mirrors the energy-pumping actions required by a human to swing a pendulum into an upright position. See **Figure 2**.

During these episodes, the DQN controller provides a rich dataset comprising sequences of visual images, pendulum states, and corresponding actions. This dataset forms the foundation for the Diffusion Policy to learn the mapping between visual observations and the appropriate control actions to balance the pendulum.

III. Model Architecture

For a visual of the full model architecture refer back to Figure 1.

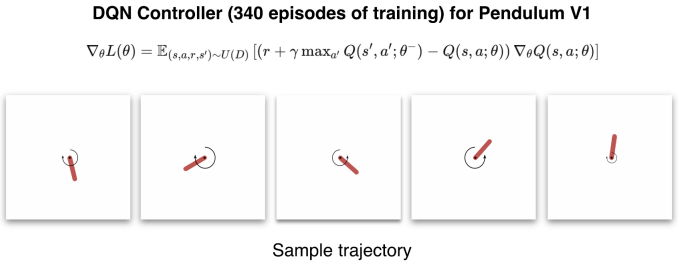

Fig. 2. Sample trajectory produced by the DQN controller.

A. Visual Encoder

The visual encoder serves as a critical component in the diffusion-based control model, providing a high-level representation of the raw visual input that can be effectively utilized for decision-making. For this purpose, I employ a ResNet (7) architecture, which has demonstrated remarkable success in various computer vision tasks due to its powerful feature extraction capabilities. Specifically, I use a ResNet variant that is pre-trained on a large dataset to leverage the rich feature representations learned from a wide array of visual contexts.

To address the temporal dynamics inherent in the pendulum control problem, I employed an approach where observations were concatenated along a temporal dimension as determined by the observation horizon parameter. This technique was crucial for maintaining a historical context of the pendulum's movements.

In this study, two distinct models were trained to explore different aspects of the problem. The first model relied solely on the visual encoder to capture both the pendulum's position and its angular velocity. This model was based on the hypothesis that the visual encoder, especially when combined with the temporal concatenation of observations, could implicitly understand the dynamics necessary for effective control, including the angular velocity, without explicitly being fed this information.

However, concerns arose about the visual encoder's ability

to simultaneously and accurately represent both the pendulum's position and angular velocity within its latent output. To address this, a second model was developed. In this variant, the latent output from the ResNet visual encoder was specifically concatenated with just the pendulum's positional data, represented by the cosine and sine of its angle. This approach placed a greater emphasis on the visual encoder to learn and represent the angular velocity dynamics based on the visual input from the observation horizon. The decision to train this second model stemmed from a desire to understand if emphasizing the pendulum's positional data would enable the ResNet to more effectively capture the necessary angular velocity dynamics for control.

B. Diffusion Model

1) Closed-loop Action Sequences: For this approach, I harness the concept of closed-loop action sequences, which allows the model to maintain temporal consistency in the actions taken while retaining the ability to respond dynamically to changes in the pendulum's state. At each time step t, the policy ingests the latest T_o steps of observation data O_t, denoted as the observation horizon. From this input, the policy predicts T_p steps of actions, where T_p is the action prediction horizon. Subsequently, T_a steps of these actions are executed, with T_a being the action execution horizon.

C. DDPM Training and Action Sequence Refinement

In the project, I adapted the Denoising Diffusion Probabilistic Model (DDPM) to meet the unique demands of controlling a pendulum based on visual inputs. This involved two key modifications. The first modification was changing the output of the DDPM to represent pendulum actions. Traditionally, the output of DDPMs is tailored for a variety of generative tasks. In this case, I reconfigured the DDPM's output to specifically represent actions within the pendulum's one-dimensional action space. Secondly, I condition the denoising process on the input observation O_t, which was derived from the visual encoder. This modification allowed the DDPM to utilize the detailed visual information processed by the encoder, making it sensitive to the specific state and dynamics of the pendulum at each time step. By embedding this observational data into the denoising process, the DDPM was empowered to generate actions that were not only denoised but also contextually relevant and accurate.

The training of the DDPM hinged on the effectiveness of the noise-prediction network, ϵ_θ. This network, essential to the denoising process, was tasked with iteratively refining the action sequences. Conditioned on the latent observation, and beginning with an initial noisy action sequence A_{t_K}, the network progressively predicted and eliminated noise from the actions through K denoising iterations. This iterative process gradually refined the action sequence, culminating in the denoised sequence A_{t_0}.

For a thorough understanding of how the DDPM is conditioned on the latent representation from the visual encoder and how the Loss is modified for denoising actionable sequences, readers are directed to the methods section of the original paper (1).

1) Inference Strategy: For inference, I utilize an action prediction horizon T_p of 8, yet only the first action of this horizon will be executed. This methodology allows for a rolling prediction window, where after the action is taken, the model is reapplied to predict the subsequent actions.

The observation model is set with an observation horizon T_o of 16, providing a compact yet informative snapshot of the pendulum's recent history. This balance is chosen to give the model sufficient context for prediction without overwhelming it with information.

D. Training

The training dataset comprised 500 episodes, each with 100 steps, generated using a DQN controller. Each step included images, the action taken, and the theta of the pendulum. Data generation utilized high-memory CPU cores on the Bigmem partition of the Harvard FAS RC cluster.

To train, I adapted the implementation code from the original Diffusion Policy paper and the PyTorch implementation of the original DDPM paper. My training script/code is available at this repo *(Note, to be able to click on hyperlinks, the pdf of this file should be downloaded from gradescope, as gradescope may not highlight hyperlinks)*. I am particularly grateful for the ReplayBuffer class (not the one used for the DQN, but the one used in the diffusion training), a critical data structure for storing demonstration datasets. The model was trained on the dataset for a few hours (approximately half a day) using two NVIDIA A100 GPUs available through the Harvard FAS RC cluster.

IV. Results

A. Initial Experiments and Observations

My initial assessment was based on the mean reward over 100 trials, with each trial comprising 100 steps. This approach, however, did not yield a clear picture of the models' capabilities due to the lack of stability in any of their performances. Therefore, I pivoted my evaluation criteria to assess the extent to which the models approximated effective control by measuring max reward per trial.

B. Experimentation with Model Configurations

In the course of the experiments, I varied observation, action, and prediction horizons and adjusted the learning parameters of the ResNet and DDPM, including their denoising schedules. However, these modifications resulted in minimal differences in the overall performance.

The initial model, which was set up with the original parameters and did not include the theta of the pendulum in its visual encoder, demonstrated notably poor performance. This model struggled significantly in stabilizing the pendulum, highlighting the limitations of the initial setup. An animation of this model's behavior for a trial is provided for reference.

C. Adaptation and Comparative Analysis

Seeking to improve the results, I modified the model by altering the information fed into the visual encoder. Specifically, I included the theta of the pendulum as part of the concatenated observations (O_t). This adjustment led to a marginal improvement in the model's performance. Interestingly, among numerous experimentation evaluation trials, one outlier exhibited similar performance to the original DQN controller (link to animation of that trial).

D. Controlled Evaluation and Insights

To gain a clearer understanding of the model's performance, I conducted a controlled evaluation using 30 different Gym environment seeds. The following plot represents the maximum reward achieved in these trials:

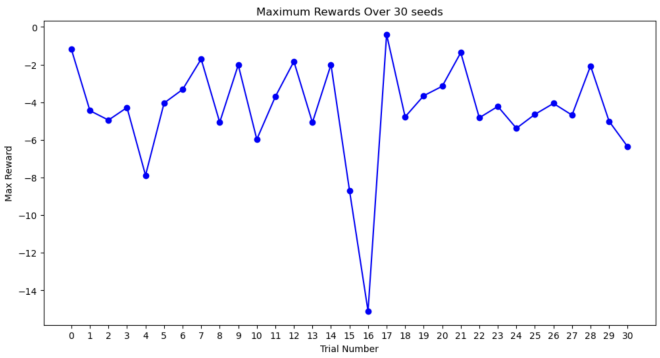

Fig. 3. Maximum Reward Achieved in Controlled Evaluation over 30 trials/seed

As indicated in the plot, the maximum reward approached zero in only one or two trials. This inconsistency underscores the model's general inability to achieve stable control of the pendulum in the upright position, for even just a few steps of each trial.

V. Discussion

A. Inability of Diffusion Policy to Achieve Stability

The reward function for the inverted pendulum problem is defined as:

$$r = -(\theta^2 + 0.1 \cdot \theta_{dt}^2 + 0.001 \cdot torque^2)$$

As said earlier, the reward function incentivizes minimal angle deviation, angular velocity, and torque application to achieve the upright position.

the experiments across 30 trials, even with the inclusion of θ in the model, revealed that the maximum reward rarely approached zero over 100 steps. This persistent deviation from the optimal reward suggests that the model struggled significantly in approximating effective control strategies. The inability to consistently achieve or even approach a state close to zero reward indicates a fundamental limitation in the model's control precision and stability to achieve the upright pendulum position or to achieve it without using excess torque application.

B. Why the Diffusion Policy Did Not Succeed?

The original implementation of the Diffusion Policy in robotics demonstrated its effectiveness in scenarios involving high-dimensional action spaces, multiple trajectories, closed-loop action sequences, and complex tasks. However, these applications were primarily goal-oriented, focusing on achieving specific objectives rather than maintaining long-term stability.

the pendulum problem is distinct in its requirement for continuous stability rather than just goal attainment. As discussed with Professor Hank Yang, stability in control problems is often a manifestation of optimal control. Therefore, any deviation from optimal control, even marginally, can significantly impact the system's stability. In this case, the Diffusion Policy, being a less-than-optimal control approximation, exhibited this very issue. Even slight inaccuracies in control decisions led to compounding instabilities, making it challenging for the model to maintain the pendulum in a stable upright position.

This phenomenon is particularly evident in this context because the inverted pendulum problem demands continuous and precise adjustments. The system's inherent instability means that any error in control can exponentially degrade the system's state. Consequently, the requirement for near-perfect control in this problem magnifies the shortcomings of the Diffusion Policy, especially in comparison to tasks where slight deviations from optimal trajectories are more permissible.

C. Computational Demands

One of the significant challenges encountered in this project was the high computational demand. The process of generating a comprehensive dataset, comprising images, pendulum states, and actions, required substantial storage capacity. Furthermore, training the DDPM on this extensive dataset necessitated the use of two NVIDIA A100 GPUs available through the Harvard FAS RC cluster.

This computational requirement starkly contrasts with more traditional control methods like Linear Quadratic Regulator (LQR) controllers or even the DQN controller, which, unlike the Diffusion Policy, do not need to use behavioural cloning for training. These traditional methods can often be implemented on devices with much lower computational resources, such as Arduinos or Raspberry Pis. Even if this method were to work, the significant computational expense associated with the approach raises questions about its practicality and scalability, especially when compared to more resource-efficient alternatives.

D. Potential Alternatives and Missed Opportunities

Reflecting on the project, there are several approaches and modifications that could have been explored before concluding on the model's efficacy:

Non-Visual Diffusion Policy: One of the alternatives worth exploring would have been the application of the Diffusion Policy using full state observations without relying on visual inputs. This could have helped isolate the impact of visual complexity on the model's performance.

Transformer Architecture for the Encoder: The original authors of the Diffusion Policy paper suggested that a transformer architecture might be more robust to problems involving frequent state changes. Implementing a transformer-based encoder could have potentially improved the model's ability to handle the dynamic nature of the pendulum control problem.

Testing on Other Stability Control Problems: It would have been insightful to evaluate the Diffusion Policy's performance on other stability-focused control problems within the OpenAI Gym framework. This could have helped determine whether the challenges faced were specific to the pendulum control problem or indicative of a broader limitation of the model in stability control scenarios.

Alternative Methods for Concatenating Observations: Exploring different strategies for concatenating observations could have provided insights into how the temporal representation of data affects the model's performance. Different concatenation methods might offer varying degrees of context and temporal understanding, which could significantly influence control outcomes.

These unexplored avenues suggest that while the current findings lean towards the limitations of the Diffusion Policy in the specific context of pendulum control, there may still be untapped potential in this approach, possibly revealed through alternative implementations or adaptations.

VI. Conclusion

This study aimed to explore the application of the Diffusion Policy to the inverted pendulum problem, a fundamental challenge in control theory. The project's hypothesis was that the Diffusion Policy's strengths in learning multimodal action distributions and solving complex vision-based manipulation tasks could be leveraged for effective control in this dynamically rich yet low-dimensional action space.

I employed a ResNet-based visual encoder and adapted the DDPM to predict actions based on visual observations. The models were evaluated on their ability to maximize rewards, indicative of control effectiveness. The results highlighted a significant challenge: despite various configurations, the models consistently failed to achieve optimal or near-optimal rewards, indicating an inability to stabilize the pendulum effectively. This outcome was contrary to the initial expectations of the Diffusion Policy's capabilities in this control scenario.

The study revealed that while the Diffusion Policy shows promise in high-dimensional, goal-oriented tasks, its application to problems requiring continuous stability, like the inverted pendulum, is less effective. This insight underscores a crucial limitation in the current scope of diffusion-based control strategies, particularly in tasks demanding precision and long-term stability.

The project also encountered substantial computational demands, raising practicality concerns, especially when compared to more traditional, resource-efficient control methods. Future research could explore alternative approaches, such as non-visual diffusion policies, transformer-based encoders, and different observation concatenation methods, to potentially enhance the model's applicability and performance in similar control tasks.

In summary, while the project did not meet its initial goal of stabilizing the pendulum using the Diffusion Policy, it provided valuable insights into the model's limitations and potential in control theory, paving the way for further research and development in this intriguing intersection of machine learning and dynamic control systems.

VII. Acknowledgements

I would like to express my sincere appreciation to Professor Hank Yang. His weekly feedback during office hours was invaluable in shaping the direction and progress of my research. Professor Yang not only introduced me to the intriguing complexities of the inverted pendulum problem during class but also provided insightful discussions about the results and implications of my project. His encouragement to pursue questions and explore the realms of control theory and machine learning, regardless of whether the results were positive or negative, was a significant source of motivation throughout this journey. I am deeply thankful for his mentorship.

I am also immensely grateful to Cheng Chi for his initial discussions regarding the feasibility of this project. His expertise in the topic and validation that the inverted pendulum problem could indeed serve as an adversarial example to the Diffusion Policy provided a strong foundation for my research.

References

[1] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song, "Diffusion policy: Visuomotor policy learning via action diffusion," 2023.

[2] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine, "Planning with diffusion for flexible behavior synthesis," 2022.

[3] Eley Ng, Ziang Liu, and Monroe Kennedy III au2, "Diffusion co-policy for synergistic human-robot collaborative tasks," 2023.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," 2020.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, "Openai gym," 2016.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Playing atari with deep reinforcement learning," 2013.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.