ESP IJSHMS
ESP International Journal of Science, Humanities and Management Studies
ISSN: 2583-9756 / Volume 3 Issue 1 January 2025 / Page No: 10-18
Paper Id: IJSHMS-V3I1P102 / Doi: 10.56472/25839756/IJSHMS-V3I1P102

Original Article

Integrating Diffusion Models into Model-Based Reinforcement Learning for Real-Time Robotic Control A Theoretical Review

Akash Vijayrao Chaudhari[1], Pallavi Ashokrao Charate[2]
[1,2] *Senior Associate, Santander Bank, Florham Park, NJ, USA, Senior Systems Analyst, Worldpay, Cincinnati, OH, USA.*

Received Date: 14 November 2024 Revised Date: 19 December 2024 Accepted Date: 05 January 2025

Abstract: *Diffusion models – a class of generative deep learning models based on iterative denoising – have emerged as powerful tools in machine learning, especially in image and sequence generation. Concurrently, model-based reinforcement learning (MBRL) has shown promise in enabling robots to plan and adapt their behavior using internal models of the environment. This review provides a comprehensive theoretical overview of recent research that integrates diffusion models into MBRL for real-time robotic control. We first summarize the foundations of diffusion models and MBRL, highlighting how diffusion's ability to model complex, multi-modal distributionsar5iv.org and MBRL's use of internal environment modelslink.springer.com can complement each other. We then survey existing methods that combine these techniques: from diffusion-based trajectory planners that treat planning as an iterative denoising processarxiv.orgdiffusion-planning.github.io, to diffusion policies that serve as powerful parametric policies in offline RL settingshuggingface.co diffusion-policy.cs.columbia.edu. The integration frameworks, their theoretical underpinnings, and key design considerations are discussed in depth. We also review use cases in robotic manipulation, locomotion, and multi-robot systems, examining how diffusion-integrated MBRL addresses real-time control challenges. Advantages of this integration – such as handling multi-modal uncertaintyarxiv.org and improving training stabilitydiffusion-policy.cs.columbia.edu – are contrasted with challenges like computational efficiency and real-world adaptation. Recent advancements (e.g., efficient diffusion sampling for faster controlarxiv.org) are highlighted, and a comparative analysis of state-of-the-art methods is presented in tabular form. Finally, we outline future directions, including opportunities to improve real-time performance, ensure safety, and combine diffusion models with other emerging paradigms. This review is intended to serve as a consolidated reference for researchers and practitioners interested in the theoretical foundations and state-of-the-art developments at the intersection of diffusion modeling and reinforcement learning in robotics.*

Keywords: *Diffusion Models, Model-Based Reinforcement Learning (MBRL), Robotic Control, Generative Modeling, Real-Time Planning, Multi-Modal Uncertainty, Trajectory Optimization, Diffusion Policies, Offline Reinforcement Learning, Robotics.*

I. INTRODUCTION

Model-based reinforcement learning (MBRL) has the potential to drastically improve sample efficiency in control tasks by learning a predictive model of the environment dynamics and using it to plan actions, as opposed to model-free methods that learn purely from trial-and-error. However, a well-known challenge is that inaccuracies in the learned model can lead to compounded errors when simulating multiple steps, causing MBRL policies to underperform compared to model-free approaches in long-horizon tasks [3]. Recently, *diffusion models* – a class of generative models known for capturing complex data distributions via iterative denoising processes – have emerged as a promising tool to improve the quality of learned dynamics models and policies in RL. Diffusion models have achieved remarkable success in domains like image generation by incrementally refining noise into realistic samples [4]. This capability to represent multi-modal distributions and generate diverse, high-fidelity outcomes has motivated researchers to integrate diffusion models into the MBRL pipeline for decision making and control.

Early work in this direction demonstrated that diffusion models can serve as powerful planners. For example, Diffuser introduced by Janner *et al.* [1] uses a diffusion model to generate future state-action trajectories, enabling an agent to plan flexible behaviors from offline data. More recently, Ding *et al.* [2] proposed Diffusion World Models (DWM), which integrate a diffusion-based dynamics model into offline RL, predicting multiple future states and rewards in a single forward pass. These studies hint that diffusion models could help overcome the limitations of traditional one-step models by providing more robust long-horizon rollouts. In parallel, other efforts have explored diffusion models for policy learning and data generation in RL, illustrating the broad potential of this integration.

This paper presents a theoretical review of existing methods that bring diffusion models into MBRL for real-time robotic control. We survey how diffusion models have been used as planners, world models, and policy generators within reinforcement learning, highlighting representative approaches (e.g., [1][2]) and their contributions. We then discuss the unique advantages conferred by diffusion modeling in an RL context, such as improved exploration and multi-modality, as well as the challenges that arise – notably computational complexity and meeting real-time constraints. Finally, we outline open research directions toward deploying diffusion-augmented MBRL in real-time robotic systems. The overall aim is to provide a cohesive understanding of this emerging interdisciplinary area and to inform future developments at the intersection of generative modeling and control.

II. BACKGROUND

A. Diffusion Models and Generative Trajectory Modeling

Diffusion models are generative models that learn data distributions by gradually perturbing data into noise and then learning to reverse this diffusion process. Originally popularized for image synthesis, diffusion models like the Denoising Diffusion Probabilistic Model (DDPM) [4] define a forward Markov chain that adds Gaussian noise to a data sample in T small steps, eventually destroying all structure into pure noise. A neural network is trained to invert this process: starting from noise, it iteratively *denoises* the sample, producing outputs that resemble the training data. After many time steps, the model can generate high-fidelity samples from scratch by following the learned reverse diffusion. Figure 1 illustrates an example of a diffusion model progressively refining noise into a clear image over multiple denoising steps.

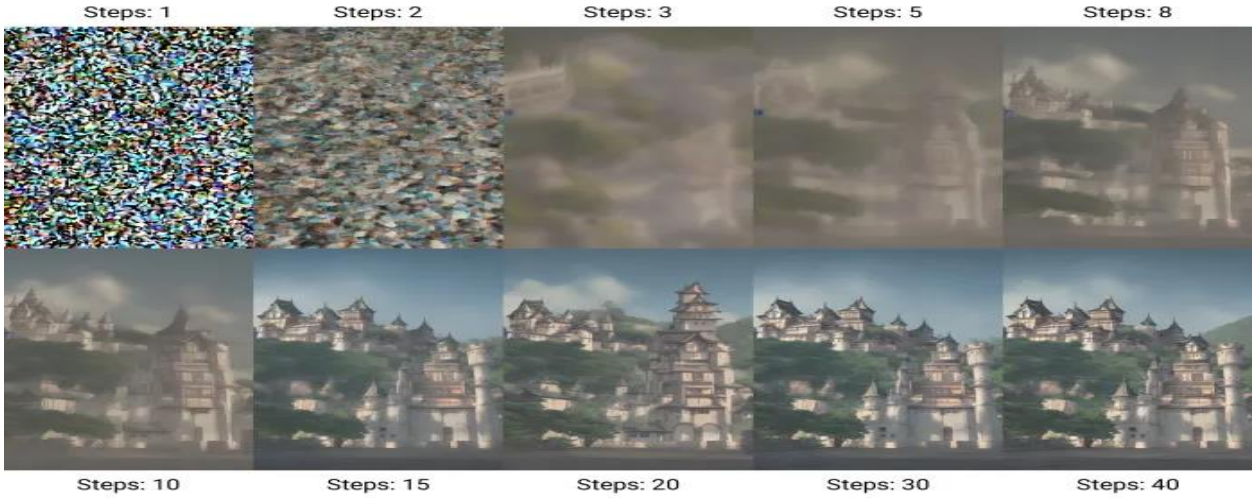

Figure 1 : Illustration of an Image Diffusion Model

Figure 1: Illustration of an image diffusion model generating a sample through iterative denoising steps (top-left to bottom-right). Starting from random noise (Step 1), the model gradually refines the image over dozens of steps until a coherent picture emerges (Step 40). Diffusion models leverage this iterative refinement to represent complex distributions with high fidelity.

Several properties make diffusion models attractive for modeling trajectories in RL. First, they can capture multi-modal distributions – for instance, multiple distinct future paths an agent might take – by learning a rich generative model rather than a single deterministic prediction. Second, diffusion models ensure temporal consistency in generated sequences through the iterative denoising, which is crucial when modeling physically plausible trajectories. Third, conditional diffusion techniques allow steering the generation process with additional inputs (often called *conditioning*). In the context of control, one can condition the diffusion model on the current state or a desired goal to generate trajectories relevant to the task. Early applications of diffusion models to sequence generation (e.g. in speech and time-series) validated their ability to handle high-dimensional, sequential data, albeit with a computational cost due to many sampling iterations.

To use diffusion models in real-time settings, a key consideration is sampling speed. Standard diffusion model sampling (with tens or hundreds of refinement steps) can be too slow for high-frequency control. Recent advances in *fast sampling* methods (e.g., DDIM and other ODE solvers for diffusion) have begun to address this by reducing the number of steps needed [4]. Additionally, classifier-guided or classifier-free guided sampling techniques allow biasing the generation toward desired

outcomes by injecting control signals (like rewards or value gradients) during the denoising process. These guidance methods have been pivotal in adapting diffusion models for decision-making tasks, as discussed later in this review. Overall, diffusion models provide a powerful *data-driven prior* over trajectories, which MBRL algorithms can leverage to imagine future outcomes beyond what simpler models (like one-step neural networks) can represent.

B. Model-Based Reinforcement Learning in Robotics

In model-based reinforcement learning, the agent builds an internal model of the environment's dynamics – often called a world model [5] – and uses it for planning or policy improvement. The world model can predict state transitions (and sometimes rewards) given the current state and action. By planning through the model (for example, via lookahead search or trajectory optimization), an agent can evaluate candidate action sequences without expensive trial-and-error in the real environment. This approach is especially useful in robotics, where each real-world interaction may be slow, risky, or costly. MBRL methods have demonstrated significantly improved sample efficiency on a variety of control problems compared to model-free methods, since they can learn from imagined experiences generated by the model [5].

A classic MBRL loop involves two main components: (a) learning the model from data, and (b) using the model for decision making. For learning dynamics, approaches range from simple analytic models (for known physics) to neural network regressors to complex latent variable models. For planning with the learned model, algorithms include shooting methods (sampling action sequences to find one with highest predicted reward), dynamic programming, or integration with trajectory optimizers (like model-predictive control). Notable examples in robotics include learning neural network dynamics for cartpole, manipulators, or locomotion and then planning using those dynamics. In principle, as the world model improves, the quality of the policy should approach that of an optimal planner with a perfect model.

In practice, however, learned dynamics models are imperfect. Model errors tend to accumulate over multi-step predictions, leading to model bias – the policy might exploit these inaccuracies in simulation in ways that do not transfer to the real environment. One manifestation is the *compounding error* problem: an error in predicting one step leads the planner into increasingly incorrect states over a long rollout, so the plan's outcome may be very poor [3]. To mitigate this, MBRL algorithms often limit planning horizon or use techniques like frequent model updates, uncertainty penalties, or mixing real and model data (as in model-based policy optimization by Janner *et al.* [3]). Ensuring model accuracy over the distribution of states induced by the policy is critical, which makes the modeling task very challenging for complex robotics domains.

Despite these challenges, MBRL has seen success in robotic control when the dynamics are sufficiently captured. Methods like Ha and Schmidhuber's World Models [5] showed that even a learned latent-space model can enable effective planning and control (e.g., driving a virtual race car using a hallucinated vision model). More recent advances such as Dreamer (Hafner *et al.*, 2019) and MuZero (Schrittwieser *et al.*, 2020) further demonstrated the power of learning world models for long-horizon tasks and games. These works set the stage for introducing even more expressive model classes – such as diffusion models – into the MBRL toolkit, with the hope of better long-term predictions and decision making.

C. Real-Time Robotic Control Considerations

When applying any RL-based controller on real robots, real-time performance is a paramount concern. Robots often operate on fast control cycles (e.g., 50–100 Hz for low-level motor control), leaving only a few milliseconds for the algorithm to compute the next action. An ideal MBRL agent for real-time control must therefore generate decisions within strict time bounds, without sacrificing the optimality of those decisions. This requirement poses a challenge for integrating complex generative models like diffusion models, which naively might require dozens of neural network evaluations per control step.

Another consideration is reliability and safety in real-time. A robot interacting with the physical world must handle uncertainties and perturbations on the fly. Any model-based planner must be robust to slight model errors or unexpected events, and ideally should come with some notion of confidence or safety margin. In safety-critical settings, violating constraints (e.g., joint limits, collision avoidance) is unacceptable, so real-time planners must incorporate constraint satisfaction mechanisms.

Existing robotic control approaches often use model-predictive control (MPC) or reactive policies because they guarantee fast responses. For MBRL methods to be viable in real-time, their planning computations might need to be streamlined or partially offloaded offline. One idea is to use a diffusion model to *pre-compute* a rich library of trajectories offline, and then at runtime quickly select or adjust one of these trajectories. Alternatively, the diffusion model's iterative refinement could be truncated for speed, or its output could warm-start an MPC solver. Ensuring that the integration of diffusion models does not break real-time requirements is an active engineering problem.

It is worth noting that the need for timely decision-making under uncertainty is not unique to robotics. For instance, in financial transaction monitoring, multi-agent RL systems have been used to detect fraud in real time [8]. In that domain, agents must react to adversarial behavior within milliseconds to prevent damage. The lessons carry over to robotics: advanced AI controllers must be both fast and adaptive. Thus, as we explore incorporating diffusion models into the control loop, we must continually assess whether the resulting system can meet the latency and robustness demands of real-world deployment. In the next sections, we review how current research is tackling these issues by marrying diffusion-based generative modeling with model-based RL techniques.

III. INTEGRATING DIFFUSION MODELS INTO MODEL-BASED RL

Integrating diffusion models into the MBRL framework can take on several forms. Broadly, researchers have explored three complementary roles for diffusion in reinforcement learning: (a) as a planner or world model for generating trajectories (replacing or augmenting conventional dynamics models), (b) as a policy representation that directly outputs actions given the state, and (c)** as a data generator to augment training data for RL. In this section, we review representative methods under these categories and discuss how they incorporate diffusion modeling into the RL loop. Table 1 provides a high-level summary of some notable approaches.

A. Diffusion Models as World Models for Planning.

One natural way to combine diffusion models with MBRL is to use the diffusion model to imagine future trajectories, which the agent can then evaluate or follow. Unlike a standard one-step predictive model, a diffusion trajectory model can produce multi-step rollouts that capture the joint distribution of an entire sequence of states and actions. The Diffuser approach by Janner *et al.* [1] pioneered this idea in offline RL. Diffuser learns a diffusion model over expert trajectories from an offline dataset. At planning time, the diffusion model generates candidate trajectories for the agent by iteratively refining noise into trajectories that both start from the agent's current state and end in high-reward outcomes. In order to bias the diffusion process toward desirable trajectories, Diffuser employs *classifier-guided sampling*: a value function or reward-conditioned classifier is used to guide the diffusion model at each denoising step, effectively steering the trajectory generation towards higher expected return paths [1]. This way, the diffusion model acts as a stochastic planner – producing a diverse set of possible futures, from which a suitable action sequence can be selected (often by taking the first action of the highest-rated generated trajectory). Empirically, Diffuser was shown to solve long-horizon tasks (like maze navigation and locomotion) from offline data as well as or better than prior planning methods, thanks to the diffusion model's ability to model multi-modal trajectory distributions (e.g., multiple ways to reach a goal).

Following Diffuser, a number of works have refined the idea of diffusion-based planners. AdaptDiffuser by Liang *et al.* [7] addresses a limitation of the original Diffuser: the diffusion model is only as good as the data it was trained on, which might not cover all scenarios or tasks of interest. AdaptDiffuser introduces an *adaptive self-evolution* mechanism where the diffusion model is iteratively improved by generating synthetic trajectories for new goals using reward gradients as guidance. In each iteration, a discriminator filters these generated trajectories for quality, and the best ones are added to the training set to finetune the diffusion model, thereby expanding its capability to handle unseen tasks. This approach effectively uses the diffusion model itself to create additional data in an online fashion, improving generalization. Results showed that AdaptDiffuser could adapt a pretrained diffusion planner to new goals (like novel target positions for a robotic arm) without needing additional expert demonstrations, significantly outperforming the original Diffuser on those new tasks [7].

Another notable extension is ensuring safety during diffusion planning. Xiao *et al.* introduced SafeDiffuser [10], which incorporates control barrier functions into the sampling procedure of a diffusion planner. The idea is to mathematically guarantee that the trajectories generated by the diffusion model satisfy safety constraints (e.g., avoiding obstacles or joint limit violations). SafeDiffuser modifies the denoising dynamics with a safety filter, so that at each step the partially generated trajectory remains within a safe set. In experiments on maze navigation and robot locomotion, SafeDiffuser was able to generate only safe trajectories while a standard diffusion planner sometimes produced unsafe ones [10]. This highlights an important direction for making diffusion planners viable in real robots: integrating domain constraints and formal guarantees.

Beyond offline settings, researchers are looking at diffusion models within online MBRL as well. Diffusion World Model (DWM) by Ding *et al.* [2] can be seen as using diffusion in the role of the dynamics predictor rather than a separate planner module. DWM is a conditional diffusion model trained to predict multiple steps of states and rewards, conditioned on the current state and a sequence of future actions (or an intended return). By predicting, say, H steps into the future in one go, DWM avoids the need to iteratively apply a one-step model H times, thus mitigating error accumulation. In their framework, the

diffusion world model was used to simulate trajectories for an offline RL algorithm (similar to model-based value expansion). The authors reported that DWM substantially improved the long-horizon prediction accuracy and achieved state-of-the-art performance on certain offline RL benchmarks, outperforming traditional one-step world models [2]. The ability of diffusion models to represent *uncertainty* over many possible futures was cited as a key factor – rather than outputting a single likely next state, the diffusion model can capture a distribution over possible outcomes, which is useful in stochastic or uncertain environments.

B. Diffusion Models as Policies for Control

Another line of integration is to use a diffusion model *in place of* a conventional policy network. In standard model-free RL or imitation learning, a policy is usually a function $\pi_\theta(a|s)$ that outputs an action distribution given the current state. Some recent works propose representing this policy implicitly via a diffusion process. One example is Diffusion Policy by Chi *et al.* [6], which learns a visuomotor policy for robotic manipulation as a conditional diffusion model. Instead of directly predicting the action for a given state (image input), Diffusion Policy trains a diffusion model that gradually denoises a random action sequence into a feasible action sequence that the robot can execute, conditioned on the observed state. During deployment, starting from random noise in the action space, the model iteratively refines the action vector such that after a fixed number of steps, the action is suitable for the current observation. This approach was shown to handle complex, high-dimensional action spaces (like torque commands for a 7-DoF arm) and multi-modal action distributions (e.g., multiple ways to grasp an object) better than traditional policy architectures [6]. Notably, Diffusion Policy achieved strong results on manipulation tasks from the Robomimic benchmark, outperforming prior imitation learning methods. The diffusion formulation provides inherent smoothing of the policy output over time (helping temporal consistency) and can naturally cover multi-modal behaviors (since diffusion can generate diverse samples).

Integrating diffusion into policy learning has also been explored in the context of offline RL with value guidance. An example is Diffusion-QL by Wang *et al.* [9], an offline RL algorithm that combines Q-learning objectives with a diffusion policy model. In Diffusion-QL, the policy is a conditional diffusion model $\pi_\theta(a_t | s_t)$ trained on offline data, but unlike pure behavior cloning, the training loss includes a term that encourages actions with higher Q-values. Essentially, they perform Q-augmented diffusion training: at each denoising step, the model is nudged to favor actions that would yield higher expected returns according to a learned critic. This can be seen as a form of classifier-free guidance where the guide is the Q-value gradient. The result is an expressive policy that is not restricted to mimic the dataset behavior but can interpolate towards better actions where supported by the Q function. Diffusion-QL was found to outperform other offline RL methods on tasks with multi-modal action distributions, demonstrating that diffusion policies can be effectively integrated into actor-critic frameworks [9].

It's worth noting that diffusion-based policies blur the line between model-based and model-free RL. On one hand, they are model-free in that they directly output actions (no explicit environment model is used at decision time). On the other hand, training such policies often leverages concepts from model-based generative modeling. In practice, diffusion policies could be combined with learned dynamics: for instance, a diffusion policy could be trained on data generated by a world model or could incorporate lookahead by conditioning on imagined future states. As of now, most diffusion-as-policy works (like [6][9]) focus on either imitation learning or offline RL settings. Applying them in online RL (where the policy must be updated continually during interaction) remains a challenge due to the computational overhead. However, one can imagine hybrid approaches – e.g., using a diffusion policy as a proposal generator that a faster policy network tracks or distills into.

C. Diffusion Models for Data Synthesis and Augmentation

A third way diffusion models contribute to RL is by generating additional data for training, effectively expanding the experience beyond what was actually collected. In offline or off-policy RL, a common problem is the limited coverage of the dataset – the agent might need to consider states or actions that are rare or absent in the data. A diffusion model learned on the dataset can serve as a *behavior prior* to sample new synthetic trajectories that resemble the real ones but with added diversity. These synthetic trajectories can be used to train value functions or policies, alleviating dataset deficiencies. For example, as mentioned above, Diffusion World Model (DWM) [2] can be viewed not only as a planner but also as a data generator: it produces fictitious trajectories into the future of the dataset, which can then be fed into an offline Q-learning algorithm (providing additional training samples that are plausible according to the world model). This is analogous to imagination-based or simulated data augmentation in model-based RL, but using a diffusion model to ensure the imagined data are realistic.

Another use case is in multi-task or meta-RL settings, where a diffusion model trained across tasks can generate task-specific data for faster adaptation. Ni *et al.* (2023) introduced MetaDiffuser, which uses a conditional diffusion model to generate

trajectories conditioned on an embedding of the task, effectively performing rapid adaptation by sampling imagined expert trajectories for a new task that can then be used to fine-tune a policy. This kind of data synthesis is powerful because the diffusion model encapsulates knowledge from many tasks and can interpolate or recombine behaviors to create novel task solutions.

In summary, diffusion models offer a versatile tool for RL: they can simulate futures (planner/world model), directly output actions (policy), or hallucinate additional experiences (data synthesizer). Each of these roles has been explored in recent literature, often with overlapping methodologies (e.g., guided sampling appears in both planning and policy contexts). Table 1 summarizes several key methods integrating diffusion models into RL, along with their primary role and domain of application.

Table 1: Representative Approaches Integrating Diffusion Models in to Reinforcement Learning.

Approach	Role of Diffusion Model	Key Idea / Application	Reference
Diffuser (2022)	Planner (trajectory generator)	Offline RL planner that generates multi-step trajectories via guided diffusion, then selects high-value actions. Demonstrated on maze and locomotion tasks.	[1]
AdaptDiffuser (2023)	Planner (self-evolving)	Diffusion planner that self-improves by generating and incorporating synthetic trajectories for new goals using reward gradient guidance. Enhances generalization to unseen tasks.	[7]
SafeDiffuser (2023)	Planner (safe trajectory gen.)	Adds safety constraints (control barrier functions) into diffusion trajectory generation to ensure all sampled plans are constraint-satisfying (collision-free, etc.).	[10]
Diffusion World Model (DWM) (2024)	World model (dynamics)	Diffusion-based predictive model that outputs multi-step state and reward sequences in one pass. Used for long-horizon rollouts in offline model-based RL, improving value estimation.	[2]
Diffusion Policy (2023)	Policy (action diffusion)	Visuomotor policy learned as a diffusion model in action space. Given the current state (image), it denoises a noise vector into an action, handling multi-modal action distributions.	[6]
Diffusion-QL (2023)	Policy (offline RL actor)	Actor-critic offline RL where the actor is a conditional diffusion model. Training includes a Q-learning term to guide the diffusion policy toward higher-value actions.	[9]

IV. CHALLENGES AND CONSIDERATIONS FOR REAL-TIME DEPLOYMENT

While the integration of diffusion models into the RL toolkit has shown considerable promise, several challenges must be addressed to deploy these techniques on real robots operating in real time. We discuss some key considerations below and potential approaches being investigated to overcome them.

A. Computational Efficiency

A standard diffusion model requires performing tens of iterative denoising steps per sample generation. In a control setting, this could mean tens of neural network forward passes to decide each action, which may be prohibitively slow for high-frequency control. Recent works have looked at accelerating diffusion sampling [4], but further optimization is needed. Options include using fewer diffusion steps (trading some optimality for speed), employing fast sampling methods like DDIM or learned diffusion samplers, or distilling the diffusion model into a simpler policy network after training. Another approach is to constrain the diffusion model architecture (e.g., smaller U-Nets or less time steps) to meet runtime budgets. For instance, Chi *et al.* [6] used only 20 denoising steps for Diffusion Policy, striking a balance between performance and latency. In hardware terms, one could leverage parallel computation on GPUs or TPUs onboard the robot to execute diffusion steps concurrently, or use model compression techniques to shrink the network. Achieving real-time control with diffusion-based planners might also involve

hierarchical planning, where a diffusion model proposes a coarse trajectory at a lower frequency and a fast local controller fills in the details at a higher frequency.

B. Integration Complexity and Stability

In an algorithm like AdaptDiffuser [7], the diffusion model is intertwined with additional components (reward gradients, discriminators, etc.). Tuning such systems can be complex. There is a risk of instability if the diffusion model generates out-of-distribution trajectories that mislead the planner or if the guidance signals (like Q-functions or reward gradients) are imperfect. Careful training procedures and regularization are required to ensure the diffusion model remains a reliable component of the control system. Moreover, when combining with traditional controllers (e.g., using a diffusion model to guide an MPC), one must ensure the two components do not conflict. A practical consideration is the memory footprint of diffusion models – they can be large, and running them alongside other control software on limited hardware might be challenging. Research into more parameter-efficient diffusion models or latent diffusion (operating on compressed representations of trajectories) could help here.

C. Safety and Reliability

As noted, ensuring that a diffusion-augmented controller behaves safely is crucial. SafeDiffuser [10] provides one template for embedding safety into the generative model. Another angle is to integrate model uncertainty awareness: if the diffusion model is uncertain (e.g., high variance in generated trajectories), the controller could default to a conservative fallback policy. Verification of such hybrid systems (diffusion model + control policy) is largely uncharted territory. We may draw on techniques from safe RL and control theory, such as online monitoring, to catch and correct any anomalous outputs from the generative model. In real robots, physical safety systems (like emergency stops) should still be in place when experimenting with these advanced controllers. Over time, as confidence in diffusion-based planners grows through testing, one can start to trust them in more safety-critical tasks.

D. Training Data Requirements

Diffusion models typically require a large amount of training data to model high-dimensional distributions effectively. For complex robotic skills, collecting a sufficiently diverse offline dataset of trajectories may be difficult. If the dataset is narrow, the diffusion model might overfit to a small subset of behaviors, limiting its utility. Techniques like AdaptDiffuser's self-generated data or leveraging simulation to real transfer (train the diffusion model on simulated data then adapt to real) can mitigate this. Another approach is to incorporate prior knowledge into the diffusion model – for example, initializing or constraining it with known physics or expert demonstrations to reduce the data needed. Federated or distributed training [8] might also be relevant if data comes from multiple sources (multiple robots or scenarios) that need to be aggregated to train a robust model.

E. Interpretability and Debugging

Diffusion models are inherently black-box generative models, which can make it hard to understand *why* a particular trajectory was suggested. In high-stakes applications, it may be valuable to have explanations for the actions chosen. Some recent work (like the use of causal inference in RL [8]) highlights the importance of interpretability. While integrating causal reasoning into a diffusion model is an open problem, simpler steps can be taken, such as analyzing the attention weights or learned score function of the diffusion model to see which parts of a trajectory it prioritizes. Visualizing generated trajectories and how they change with guidance inputs (e.g., varying the strength of reward guidance) can also give insights into the model's decision-making process. For deployment, one might implement monitors that check the diffusion model's output against known logical constraints or expected behavior patterns as a sanity check.

In summary, bridging the gap between the algorithmic advances in diffusion-based RL planning and the practical demands of real-time robotic control will require innovations on multiple fronts: algorithmic speed-ups, system integration engineering, safety assurance, and clever use of data. The challenges are significant but surmountable. As hardware accelerators improve and research continues to streamline diffusion models, we expect the gap to narrow. The next section outlines some future directions that could further facilitate the use of diffusion models in everyday robot learning and control.

V. CONCLUSION AND FUTURE DIRECTIONS

The incorporation of diffusion generative models into model-based reinforcement learning represents a novel convergence of sequence modeling and decision making, and it has opened up promising avenues for improving robotic control. This theoretical review has surveyed the state-of-the-art in this young but rapidly evolving area. Diffusion models have been successfully used as trajectory planners [1], as world models for value optimization [2], and as expressive policy representations

[6] in various RL settings. Across these works, a common theme is that diffusion models help capture the rich, multi-modal nature of real-world decision processes – whether it's the many possible ways to complete a task or the uncertainty inherent in a robot's interactions. By leveraging this strength, MBRL algorithms augmented with diffusion models can achieve a level of foresight and flexibility that is difficult to attain with conventional dynamics models or policies.

Despite the progress, the journey toward *real-time* diffusion-based control in robots is just beginning. A number of important future research directions can be identified:

A. Model Distillation and Hierarchical Control

To meet real-time constraints, one promising direction is distilling the knowledge of a diffusion planner into a simpler policy (e.g., a neural network or a low-dimensional controller). After using diffusion to explore the policy space during training, one could train a secondary policy to imitate the diffusion model's decisions without needing iterative sampling. Alternatively, a hierarchy could be used where the diffusion model operates at a high level (proposing sub-goals or coarse reference trajectories at a lower frequency) and a fast low-level controller handles instantaneous control. This would retain the benefits of diffusion-generated global plans while keeping the real-time control loop lightweight.

B. Improved Sampling Algorithms

Ongoing research in the diffusion model community on faster ODE/SDE solvers, partial sampling, and learned samplers will directly benefit the RL integration. For instance, methods that learn to jump directly to late diffusion steps (thus requiring fewer iterations) or that adaptively allocate more refinement to critical time regions of the trajectory could reduce computation. In RL, one could exploit the fact that subsequent actions in a trajectory are highly correlated – perhaps warm-starting the diffusion chain for the next step using the result from the previous step, rather than starting from pure noise each time. This temporal coherence could be harnessed to cut down redundant computation.

C. Multi-Agent and Multi-Modal Interactions

Thus far, most diffusion-RL research has focused on single-agent scenarios. Extending these ideas to multi-agent environments (where multiple robots or agents are interacting) is a fertile ground. A diffusion model could, for example, jointly generate trajectories for multiple agents, implicitly learning their coordination patterns. Some initial work in this direction (e.g., MADiff for multi-agent diffusion [9]) indicates diffusion models can capture the joint behavior of multiple agents. We may see diffusion-enabled multi-robot planners that can negotiate and co-plan in shared spaces. Additionally, integrating diffusion models with other modalities (such as natural language or vision goals) could enable *instruction-following* robots: e.g., a diffusion policy that conditions not just on the current state but also on a language command to generate a compliant action sequence.

D. Benchmarking and Real-World Trials

As methods mature, a critical step will be rigorous benchmarking in both simulation and on real robotic hardware. Standardized tasks (perhaps extensions of existing benchmarks like D4RL or Robomimic to include timing and safety metrics) will help quantify the benefits of diffusion integration. Real-world trials, even in simplified form, are essential to uncovering unmodeled challenges (such as sensor noise or actuator delays) that might not appear in simulation. Early adoption in lower-risk scenarios – for instance, diffusion-guided trajectory planning for robot manipulators in controlled industrial settings – could demonstrate viability and build trust in the approach.

E. Theoretical Understanding

On the theoretical side, questions remain about why and when diffusion models provide the biggest gains in RL. Is it primarily their multi-modal generative ability, or do they regularize the policy/model learning in some beneficial way (e.g., through the stochastic smoothing of the diffusion process)? Developing a theory of how the diffusion prior interacts with value optimization could guide the design of even better algorithms. Moreover, understanding the convergence properties and stability of diffusion-in-the-loop systems (perhaps via dynamical systems theory or probabilistic inference perspectives) would add confidence for deployment.

In conclusion, integrating diffusion models into model-based RL is a compelling paradigm that leverages the best of two worlds: the creativity and expressiveness of generative models, and the goal-directed adaptability of reinforcement learning. The approaches reviewed in this paper have already provided proof-of-concept results that diffusion-driven planners and policies can excel in complex control tasks. By continuing to refine these methods and address the outlined challenges, we move closer to a future where robots can plan, learn, and act in real time with a powerful generative imagination at their core. Such robots would

not only react to the present, but also *anticipate the future* in rich, probabilistic detail – enabling more intelligent and safe autonomy in unpredictable environments.

VI. REFERENCES

[1] Chaudhari, A. V. (2025). AI-powered alternative credit scoring platform. ResearchGate. https://doi.org/10.13140/RG.2.2.13191.92325

[2] Chaudhari, A. V. (2025). A cloud-native unified platform for real-time fraud detection. ResearchGate. https://doi.org/10.13140/RG.2.2.19902.80962

[3] Chaudhari, A. V., & Charate, P. A. (2024). Data Warehousing for IoT Analytics. International Research Journal of Engineering and Technology (IRJET), 11(6), 311–320

[4] Chaudhari, A. V., & Charate, P. A. (2025). AI-Driven Data Warehousing in Real-Time Business Intelligence: A Framework for Automated ETL, Predictive Analytics, and Cloud Integration, International Journal of Research Culture Society (IJRCS), 9(3), 185–189

[5] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," arXiv preprint arXiv:1803.10122, 2018.

[6] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion Policy: Visuomotor policy learning via action diffusion," arXiv preprint arXiv:2303.04137, 2023.

[7] Z. Liang, Y. Mu, M. Ding, F. Ni, M. Tomizuka, and P. Luo, "AdaptDiffuser: Diffusion models as adaptive self-evolving planners," in Proc. of ICML, 2023.

[8] A. V. Chaudhari and P. A. Charate, "Autonomous AI agents for real-time financial transaction monitoring and anomaly resolution using multi-agent reinforcement learning and explainable causal inference," International Journal of Advance Research, Ideas and Innovations in Technology, vol. 11, no. 2, 2025.

[9] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion-QL: Diffusion policies as an expressive policy class for offline reinforcement learning," in Proc. of ICLR, 2023.

[10] W. Xiao, T. H. Wang, C. Gan, R. Hasani, M. Lechner, and D. Rus, "SafeDiffuser: Safe planning with diffusion probabilistic models," in Proc. of ICLR, 2025 (to appear).