13th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN), Silver Lake, 8-11 June 2026

Flow Matching Policy for Behavioral Cloning

Mihailo Radović
Applied Mathematics
University of Belgrade, School of Electrical Engineering
Belgrade, Serbia
rm253228m@student.etf.bg.ac.rs

Filip Marčić
Applied Mathematics
University of Belgrade, School of Electrical Engineering
Belgrade, Serbia
mf253318m@student.etf.bg.ac.rs

***Abstract*—Behavioral cloning (BC) is a foundational imitation learning paradigm, but many standard continuous-control BC baselines rely on unimodal Gaussian policies or other relatively low-expressivity action parameterizations. Consequently, they struggle to capture the complex, multi-modal strategies present in diverse offline datasets, such as those containing human, medium-quality, or mixed trajectories, leading to a significant performance gap. To address this limitation, we introduce the Flow Matching Policy (FMP), a highly expressive representation for continuous control BC. Our approach models the conditional action distribution as a continuous-time normalizing flow, learning an observation-conditioned velocity field to transport a simple base noise distribution into the empirical action distribution. Evaluations against strong Gaussian and diffusion policy baselines across standard continuous control benchmarks demonstrate that the FMP consistently achieves competitive or superior performance. These results suggest that continuous-time flow models are a promising alternative for capturing highly complex and varied behaviors from noisy data.**

Keywords—behavioral cloning, flow matching policy, Gaussian policy, diffusion policy

I. Introduction

Machine learning and robotics applications are often constrained by limited datasets, typically consisting of human-generated demonstrations of varying quality. Despite these challenges, behavioral cloning (BC) serves as a common and effective baseline, training a policy in a supervised manner using state-action pairs [1], [2]. In continuous control settings, these policies are often modeled using Gaussian distributions. While this approach is simple and computationally efficient, it suffers from fundamental limitations in environments where multiple distinct actions may be optimal for a given state. In such cases, unimodal Gaussian policies tend to average across possible actions, often resulting in suboptimal or unsafe behavior.

As an alternative, recent advances in generative modeling have introduced methods based on differential equations for modeling continuous data distributions [3]. Building on these approaches, this work investigates the Flow Matching Policy (FMP) as an alternative to the Gaussian formulation. Flow Matching provides a framework for learning transformations from simple noise to complex data distributions [4]. By formulating the action generation process as a continuous-time vector field regression problem, this approach enables greater policy expressivity and potentially improves the modeling of complex, multimodal action spaces.

To investigate this, we evaluate both approaches across an array of continuous control benchmarks, including tasks from the MuJoCo [5] and D4RL suites [6]. We systematically compare the FMP against strong Gaussian and diffusion policy baselines to observe their respective behaviors in environments with varying levels of complexity and data quality. Our results indicate that the FMP offers a viable alternative to the standard baseline, showing particular utility in environments characterized by multimodal or mixed-quality data, such as the D4RL Pen, AntMaze and Kitchen tasks. In these settings, the flow-based model mitigates the action-averaging issue often observed in unimodal baselines. However, we also note that this expressivity comes at the cost of increased computational and methodological complexity. Ultimately, this work adapts the flow matching framework for continuous-control BC and provides a comparative analysis to better understand the trade-offs between simple Gaussian formulations, diffusion policies, and continuous-time normalizing flows in offline imitation learning.

II. Related Work

A. Diffusion-Based Behavioral Cloning

Recent advancements in offline imitation learning have heavily leveraged diffusion models to overcome the limitations of explicit likelihood models. Methods such as Behavioral Cloning via Diffusion [7] and Diffusion Policy [8] frame action generation as a sequential denoising process, allowing the policy to capture highly multimodal and asymmetric data distributions without relying on mode-seeking objectives. While these diffusion-based policies achieve strong performance on complex manipulation and locomotion tasks, their reliance on iterative Langevin dynamics or denoising diffusion probabilistic models (DDPM) [9] introduces significant inference latency. Our flow matching approach serves as a natural evolution of this line of work, retaining the ability to model arbitrary action distributions while replacing the computationally costly reverse diffusion process with a highly efficient, deterministic ordinary differential equation (ODE) solver.

B. Continuous Normalizing Flows and Neural ODEs

The theoretical foundation of our approach is rooted in Continuous Normalizing Flows (CNFs) and Neural Ordinary Differential Equations [3]. Traditionally, training CNFs required computationally expensive simulations through ODE solvers to compute exact log-likelihoods [10], making them difficult to scale for high-frequency control tasks. The recent introduction of Flow Matching [4] and Rectified Flows [11] bypassed this limitation by framing the training process as a simulation-free vector field regression task. By adapting this framework to observation-conditioned action generation, our method inherits the theoretical guarantees of CNFs while avoiding prohibitive training overhead.

C. Flow Matching Policy Gradients

Recent literature has begun exploring continuous-time normalizing flows for decision-making, notably through

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Accepted version. Published in: 2026 13th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), pp. 1–6, DOI: 10.1109/ICETRAN71739.2026.11727690. Final version: https://ieeexplore.ieee.org/abstract/document/11727690.

methods like Flow Matching Policy Gradients (FMPG) and related flow-based actor-critic frameworks [12]. These approaches integrate flow-based generative models into an online reinforcement learning setting, allowing agents to iteratively explore environments and refine their policies based on reward signals. While this demonstrates the theoretical viability of vector field regression for continuous control, it focuses heavily on the complexities of online exploration and value estimation. In contrast, our work isolates the generative capability of flow matching, applying it strictly within an offline, supervised behavioral cloning setting to directly evaluate its capacity for modeling fixed datasets.

III. Flow Matching Policies

A. Gaussian Policies and Their Limitations

In standard continuous-control behavioral cloning (BC), the policy is typically parameterized as a conditional Gaussian distribution [1]. A neural network processes the state to output the mean vector and the covariance matrix. The model is trained by minimizing the negative log-likelihood of the expert actions.

While computationally efficient and simple to implement, Gaussian policies suffer from a fundamental limitation: they are strictly unimodal. When trained on diverse or mixed-quality datasets, where an expert might demonstrate multiple distinct, equally valid ways to navigate a state, the maximum likelihood objective forces the Gaussian to fit all modes simultaneously. This leads to mode-averaging, where the policy predicts the mean of the possible actions, often resulting in suboptimal or catastrophic behavior [13], [14]. To model highly expressive, multimodal expert behaviors, a paradigm shift from simple likelihood maximization to generative sequence modeling is required.

B. Flow Matching

Flow Matching (FM) [4] provides a simulation-free framework for training Continuous Normalizing Flows (CNFs) [10] to model complex data distributions. The core intuition is to construct a continuous-time transformation that gradually reshapes a simple base noise distribution (e.g., standard normal) into a target empirical data distribution. This transformation is governed by an ODE defined by a time-dependent vector field:

$$\frac{\mathrm{dz}}{\mathrm{dt}} = v(z,t), \quad t \in [0,1], \tag{1}$$

where z is the point being transformed, $t \in [0,1]$ is time, and $v(z, t)$ is the vector field. Instead of simulating the ODE during training to compute exact log-likelihoods, which requires prohibitive computational overhead [10], FM constructs a known target vector field based on marginal probability paths. By defining a simple linear interpolation between a noise sample and a data sample, the network can be trained via direct regression to predict the trajectory's velocity, entirely bypassing the need for ODE solvers during the backward pass [11].

C. Proposed Flow Matching Policy

Building upon this generative framework, we propose a FMP tailored for offline continuous control. We formulate action generation as the integration of a state-conditioned vector field. To capture environmental context, the state is processed by a multi-layer perceptron (MLP) feature extractor to produce a hidden representation. The continuous integration time is embedded using high-frequency sinusoidal positional encodings followed by an MLP, similar to the representations utilized in GLIDE [15]. This ensures the network is highly sensitive to the temporal dynamics of the ODE.

We sample the base noise from a Gaussian distribution. Because the FM formulation operates optimally in unbounded spaces, we map the bounded expert actions to an unconstrained latent space using the inverse hyperbolic tangent, resulting in the target. We define a linear, constant-velocity interpolation path:

$$z_t = (1-t)z_0 + tz_1, \tag{2}$$

where z_0 is the starting point, z_1 is the target datapoint, $t \in [0,1]$ is the interpolation time, and z_t is the point along the linear path from z_0 to z_1 at time t. The target velocity along this path is the time derivative of z_t. We parameterize a neural velocity field to predict this target velocity, given the current interpolated point, the state representation, and the time step. The resulting BC objective is the mean squared error between the predicted and target velocities:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbb{t},\mathbb{z}_0,\mathbb{z}_1}[\omega(t)\, \|v_\theta(z_t, h(s), t) - (z_1 - z_0)\|_2^2]. \tag{3}$$

Here, θ represents the parameters of the neural velocity field. The expectation is taken over $t \sim p(t)$, initial noise $z_0 \sim \mathcal{N}(0, I)$, and empirical expert actions $z_1 \sim \mathcal{D}$ (mapped to the unbounded latent space). The variable z_t is the linearly interpolated point at time t, and the target velocity is the constant vector $(z_1 - z_0)$.

To optimize the network, the time variable t is sampled from a Logit-Normal distribution [16], $p(t)$, which concentrates sampling density around $t = 0.5$. This intentionally biases the sampling toward intermediate time steps, as the geometry of the vector field exhibits the highest curvature and is most challenging to learn during the middle stages of the transport process. Furthermore, $\omega(t)$ acts as a temporal weighting function. Since gradients can become unstable near the boundary conditions ($t \to 0$ and $t \to 1$), applying this parabolic weighting scheme down-weights the loss at the extremes, ensuring stable and robust optimization throughout training.

During training, we maintain an Exponential Moving Average (EMA) of the model weights. The EMA significantly smooths the learned vector field and prevents training collapse, a technique proven critical for stabilizing continuous-time generative models [4], [9].

During evaluation, the action is generated by numerically simulating the learned ODE using the Euler method. To ensure stable and deterministic evaluation, a standard requirement for continuous control benchmarks, we fix the base noise at the origin (i.e., initialize it as a zero tensor). The velocity field is integrated over a fixed number of discrete equidistant steps:

$$z_1 = z_0 + \int_0^1 v_\theta(z_t, h(s), t)\mathrm{d}t, \tag{4}$$

where z_1 is the updated state, z_0 is the initial state, t is the time variable, and v_θ is the learned vector field that determines how z evolves over time. The final environmental action is obtained by squashing the integrated variable back into the valid action range via $a = tanh(z_1)$.

Our approach can be contrasted with recent developments in online reinforcement learning, such as FMPG [12]. While FMPG utilizes a continuous-time flow for action generation, it optimizes the vector field dynamically using a Q-function critic to maximize expected future rewards through active environment interaction. In contrast, our formulation isolates the generative capacity of the flow model strictly for offline imitation learning. By directly regressing the empirical expert vector field, our policy avoids the instability of temporal difference learning and value overestimation, providing a simulation-free training paradigm for extracting expressive behaviors from fixed datasets. The implementation is publicly available at: https://github.com/mradovic38/flow-bc.

IV. Comparative Analysis of Results

The results presented in Table 1 demonstrate a performance comparison between the unimodal Gaussian baseline, diffusion policy, and the proposed FMP. The diffusion policy baseline was implemented using DDIM-based [17] sampling with a DDPM [9] training objective, sharing the same backbone architecture as the Gaussian and FMP models to ensure a fair comparison. Our FMP demonstrates strong performance, frequently outperforming both baselines, particularly in MuJoCo environments requiring complex coordination (e.g., Humanoid) or D4RL tasks with highly multimodal data (e.g., Kitchen, AntMaze).

A. MuJoCo Benchmarks

In the standard MuJoCo locomotion tasks [5], displayed in Fig. 1, the FMP matches or exceeds the Gaussian and diffusion policy baselines in four out of six environments. Notably, in Humanoid environment, which represents the highest state-action dimensionality in the suite, the FMP achieves a significantly higher mean return (8213.03) with a substantially reduced standard deviation compared to the Gaussian and diffusion policies (36.86 compared to 1533..97 and 1838.37). This suggests that the continuous-time formulation of the FMP provides a more stable integration path for complex coordination tasks where minor deviations in action selection can lead to catastrophic balance loss. Conversely, in simpler environments like Swimmer and Walker2d, the Gaussian baseline remains highly competitive. We attribute this to the unimodal nature of the medium datasets for these specific tasks; when the expert data are concentrated around a single trajectory, the added expressivity of a flow-based model introduces unnecessary complexity without a corresponding performance gain.

TABLE I. Performance Comparison on MuJoCo and D4RL Tasks (Average Return over 100 Evaluation Episodes ± Standard Deviation).

Environment	*Dataset*	*Gaussian Policy*	*Diffusion Policy*	*FMP (ours)*
MuJoCo				
HalfCheetah	medium-v0	14871.47 ± 3046.86	15081.97 ± 1440.60	**15499.46 ± 51.01**
Hopper	medium-v0	3577.98 ± 29.51	3243.76 ± 656.69	**3593.45 ± 31.59**
Humanoid	medium-v0	7581.16 ± 1838.37	7775.93 ± 1533.97	**8213.03 ± 36.86**
Swimmer	medium-v0	**274.17 ± 19.95**	220.11 ± 3.67	227.50 ± 12.08
Walker2d	medium-v0	**6235.86 ± 29.79**	6150.82 ± 106.45	6204.74 ± 79.81
Ant	medium-v0	5769.32 ± 1602.93	5931.90 ± 1272.18	**6027.36 ± 1050.83**
D4RL				
Door	human-v2	158.00. ± 203.86	**314.26 ± 323.47**	279.99 ± 327.75
Pen	human-v2	2636.42 ± 4007.15	5143.41 ± 4304.77	**5410.04 ± 4336.66**
Kitchen	mixed-v2	603.38 ± 234.78	500.84 ± 194.93	**740.16 ± 151.83**
AntMaze	medium-play-v1	0 ± 0.00	12.83 ± 64.96	**32.09 ± 104.37**

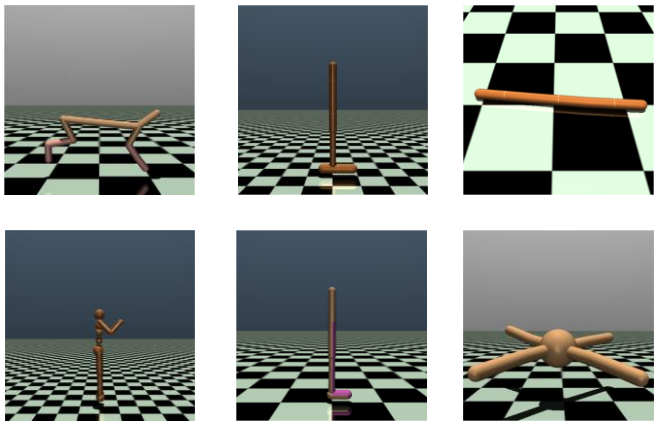


Fig. 1. Tested MuJoCo environments: HalfCheetah (top left), Hopper (top middle), Swimmer (top right), Humanoid (bottom left), Walker2d (bottom middle), Ant (bottom right).

B. D4RL Benchmarks

The most significant advantages of the FMP are observed in the D4RL suite [6], particularly in the Kitchen, Pen and AntMaze tasks, displayed in Fig. 2. The datasets used in Door, Kitchen and Pen tasks (human-v2 and mixed-v2) are intentionally designed to be non-ideal, some containing sequences of human demonstrations that are inherently multimodal. For example, the mixed dataset for the Kitchen environment requires an agent to complete four sub-tasks out of seven possible interactions (e.g., toggling a switch, opening a microwave, moving a kettle, or sliding a cabinet). Because the dataset contains trajectories collected from policies at different stages of training, including both suboptimal and near-optimal behaviors, the ordering of sub-task completion varies considerably across trajectories. One policy might open the microwave before moving the kettle, while another does the reverse. Consequently, the action distribution for a given observation in intermediate kitchen states is highly multi-modal. On the other hand, AntMaze introduces sparse-reward navigation in a maze environment with structured but heterogeneous trajectory collections. In the medium-play setting, the dataset contains trajectories generated by partially trained policies exhibiting varying navigation strategies and path choices, resulting in a highly multimodal action distribution. Multiple distinct trajectories may successfully reach the same

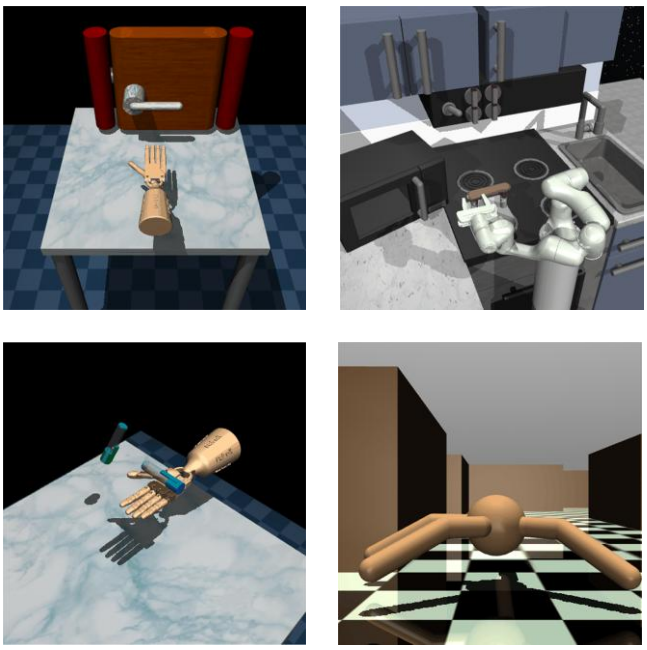


Fig. 2. Tested D4RL environments: Door (top left), Kitchen (top right), Pen (bottom left), AntMaze (bottom right).

goal state, requiring the policy to model diverse long-horizon behaviors rather than collapsing toward averaged actions that fail to complete the task In this setting, Gaussian policies, diffusion policies, and FMP exhibit distinct trade-offs in performance and modeling capacity:

- **Mode averaging and dataset imbalance:** The Gaussian policy's lower performance in D4RL tasks is a classic symptom of mode averaging. Because a unimodal Gaussian must assign high probability to the mean of all expert actions, it often predicts an "average" action that corresponds to no valid expert behavior [8]. Diffusion policy does not explicitly average modes in the same way; however, its lower performance in this setting may be attributed to sensitivity to the highly imbalanced and heterogeneous trajectory distribution, where rare successful sub-task orderings are harder to consistently recover. FMP, by directly modeling structured vector fields over trajectories, more effectively concentrates on coherent multi-step sub-task sequences, resulting in higher returns.
- **Manipulation tasks (Door and Pen):** In the Door task, diffusion policy achieves the highest performance, outperforming FMP and Gaussian policy. In the Pen task, both diffusion policy and FMP significantly outperform Gaussian policy, with FMP achieving the strongest mean return. Across both tasks, all methods exhibit high variance, reflecting the multimodal and stochastic nature of the human demonstration dataset. However, diffusion policy and FMP are both able to recover high-return trajectories more frequently than the Gaussian policy, which tends to average across inconsistent action modes.
- **AntMaze long-horizon navigation:** A more pronounced distinction emerges in AntMaze. While the Gaussian policy fails to make meaningful progress, both diffusion policy and FMP are able to solve the task to some extent, with increasing performance from Gaussian to diffusion to FMP. Specifically, diffusion policy achieves modest returns, while FMP attains the highest average performance in the medium-play-v1 setting. However, all methods exhibit high variance, reflecting the difficulty of sparse-reward exploration and the stochasticity of long-horizon trajectory rollouts. These results suggest that FMP may offer improved performance in sparse-reward navigation tasks, where maintaining coherent long-term behavior is important.

C. Ablation Study on Integration Steps

A defining characteristic of the FMP is the requirement to numerically solve the ordinary differential equation (ODE) during inference. To quantify the trade-off between policy performance and computational latency, we conducted an ablation study on the number of Euler integration steps (N) utilized during evaluation. We selected Humanoid-medium-v0 to represent high-dimensional locomotion stability and Kitchen-mixed-v2 to represent highly multimodal, multi-task behavior. To establish a reproducible baseline for CPU-bound robotic control, inference latency was measured using a batch size of 1 on a single thread of an Apple M4 Pro processor with 48GB of unified memory.

As shown in the ablation data, in Table 2, simulating the vector field with only $N = 1$ or $N = 2$ steps resulted in near-total failure in the Kitchen task and sub-optimal performance in Humanoid. This confirms that the learned mapping from noise to the empirical action distribution is highly non-linear and necessitates continuous integration. Performance improved sharply as the step count increased, though the optimal integration resolution varied by environment complexity. For the high-dimensional continuous control required in Humanoid, the FMP achieved optimal stability at $N = 12$ steps, marked by the highest mean return and a dramatically reduced standard deviation. Conversely, the highly multimodal and noisy Kitchen dataset required a slightly higher integration resolution, achieving peak performance and mode-separation at $N = 16$ steps.

Crucially, increasing N beyond these optimal thresholds (to 24, 36, or 50 steps) yielded no meaningful performance gains, while inference latency scaled linearly. In Humanoid environment, at $N = 12$, the FMP requires 0.78 ms to generate an action, whereas $N = 50$ increases the latency to 3.09 ms without corresponding behavioral improvements. This empirical analysis confirms that a remarkably low number of Euler steps (12 to 16) offers an optimal balance, achieving the expressivity required for complex control while minimizing the accumulation of inference latency inherent to continuous-time flow models.

V. Conclusion, Limitations and Future Work

The empirical evaluation conducted in this work demonstrates that FM policies represent a robust and expressive alternative to traditional Gaussian formulations in BC. By replacing the restrictive unimodal likelihood objective with a continuous-time vector field regression, alternative to traditional Gaussian formulations in BC. By replacing the restrictive unimodal likelihood objective with a continuous-time vector field regression, the FMP successfully captures the diverse behaviors present in non-ideal datasets, such as the human-

TABLE II. ABLATION OF INTEGRATION STEPS ON PERFORMANCE AND INFERENCE LATENCY (AVERAGE RETURN OVER 100 EVALUATION EPISODES ± STANDARD DEVIATION; LATENCY IN MILLISECONDS).

ODE steps (N)	*Latency (ms)*	*Average return ± std*
Humanoid-medium-v0		
1	0.11	7031.33 ± 2263.08
2	0.16	7642.93 ± 1824.08
4	0.29	7957.36 ± 1214.06
8	0.53	7836.65 ± 1445.65
10	0.65	8172.81 ± 574.24
12	0.78	**8213.03 ± 36.86**
16	1.03	7871.77 ± 1317.44
20	1.26	7979.39 ± 1233.45
24	1.50	7973.92 ± 1186.77
36	2.21	8061.47 ± 967.44
50	3.09	8034.26 ± 1178.72
Kitchen-mixed-v2		
1	0.13	17.35 ± 61.87
2	0.20	17.78 ± 78.61
4	0.34	594.98 ± 236.30
8	0.62	674.19 ± 161.76
10	0.75	611.45 ± 286.67
12	0.91	539.10 ± 295.77
16	1.18	**740.16 ± 151.83**
20	1.45	681.41 ± 213.52
24	1.71	707.02 ± 206.12
36	2.58	722.46 ± 172.70
50	3.51	730.37 ± 166.26

collected and mixed-quality trajectories found in the D4RL suite [6]. The mitigation of mode-averaging in tasks like Kitchen and AntMaze suggests that the generative capacity of FM-based models is particularly well-suited for high-dimensional control problems where expert behavior is inherently varied and inconsistent [13].

Furthermore, the stability observed in high-dimensional MuJoCo tasks such as Humanoid [5] indicates that the integration of a learned velocity field provides a reliable control signal. Despite these advantages, the added expressivity of the flow matching framework involves specific methodological trade-offs. The most prominent limitation is the computational cost associated with numerical ODE integration during inference. Unlike the single-pass nature of Gaussian policies, the FM policy requires multiple sequential network evaluations to solve the ODE; while we found that a limited number of Euler steps (10 to 20) was sufficient for our benchmarks, this added latency may be prohibitive for ultra-high-frequency real-time robotics applications. Additionally, our results for structurally simpler tasks, such as Swimmer and Walker2d, suggest that for strictly unimodal or low-entropy datasets, the FMP may introduce unnecessary complexity without yielding a significant performance gain. In such cases, the computationally efficient Gaussian baseline remains a highly competitive and perhaps preferable choice. Moreover, an extensive investigation into the relationship between environment entropy and the performance gap of FM-based policies will be essential to better characterize when the benefits of generative expressivity outweigh the costs of computational overhead. Expanding these evaluations to long-horizon manipulation and multi-agent coordination remains a critical area for validating the broader utility of continuous-time normalizing flows in embodied AI.

REFERENCES

[1] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," Advances in Neural Information Processing Systems, vol. 1, 1988.

[2] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in Proceedings of the Fourteenth International Conference on Machine Learning (ICML), pp. 12–20, 1997.

[3] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," Advances in Neural Information Processing Systems, vol. 31, 2018. URL: https://arxiv.org/abs/1806.07366

[4] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in International Conference on Learning Representations (ICLR), 2023. URL: https://arxiv.org/abs/2210.02747

[5] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033, 2012.

[6] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: Datasets for deep data-driven reinforcement learning," arXiv preprint arXiv:2004.07219, 2021. URL: https://arxiv.org/abs/2004.07219

[7] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Devlin, N. Radin, and K. Hofmann, "Imitating human behaviour with diffusion models," in International Conference on Learning Representations (ICLR), 2023. URL: https://arxiv.org/abs/2301.10677

[8] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in Proceedings of Robotics: Science and Systems (RSS), 2023. URL: https://arxiv.org/abs/2303.04137

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in Neural Information Processing Systems, vol. 33, pp. 6840–6851, 2020. URL: https://arxiv.org/abs/2006.11239

[10] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "FFJORD: Free-form continuous dynamics for scalable reversible generative models," in International Conference on Learning Representations (ICLR), 2019. URL: https://arxiv.org/abs/1810.01367

[11] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in International Conference on Learning Representations (ICLR), 2023. URL: https://arxiv.org/abs/2209.03003

[12] D. McAllister, S. Ge, B. Yi, C. M. Kim, E. Weber, H. Choi, H. Feng, and A. Kanazawa, "Flow matching policy gradients," unpublished.

[13] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in Proceedings of the 5th Conference on Robot Learning (CoRL), vol. 164, pp. 1678–1690, 2022.

[14] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in Proceedings of the 5th Conference on Robot Learning (CoRL), vol. 164, pp. 158–168, 2022. URL: https://arxiv.org/abs/2109.00137

[15] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," in International Conference on Machine Learning (ICML), pp. 16784–16804, 2022. URL: https://arxiv.org/abs/2112.10741

[16] P. Esser, S. Kullu, A. Stevens, R. Beaumont, A. Sauer, C. Schuhmann, and R. Rombach, “Scaling rectified flow transformers for high-resolution image synthesis,” arXiv preprint arXiv:2403.03206, 2024. URL: https://arxiv.org/abs/2403.03206

[17] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in International Conference on Learning Representations (ICLR), 2021. URL: https://arxiv.org/abs/2010.02502