Robot Manipulation with Flow Matching

Fan Zhang, Michael Gienger
Honda Research Institute EU
fan.zhang@honda-ri.de

Abstract: This paper presents a new imitation learning paradigm for robot manipulation with flow matching policy. Flow matching represents a robot visuomotor policy as a conditional process of flowing random waypoints to desired robot action trajectories, by regressing vector fields of fixed conditional probability paths. We evaluate the proposed method across two simulation benchmarks and a real-world dataset with 10 tasks across Activities of Daily Living. Our extensive evaluation highlights that learning multimodal robot actions with flow matching policy leads to consistently more stable training and faster generalization than alternative diffusion-based behavior cloning methods. https://hri-eu.github.io/flow-matching-policy/.

Keywords: Flow Matching Policy, Robot Manipulation

1 Introduction

From the traditional behavior cloning with convolutional networks [1] to transformer-based learning structures [2], extensive research has modeled robot action trajectories from visual scenes. A recent line of works builds on successes in diffusion models [3] to generate motion trajectories to capture multimodal action distributions. Flow Matching is another novel generative method. Sharing theoretical similarities with stochastic Denoising Diffusion Probabilistic Models, flow matching aims to regress onto a deterministic vector field to flow samples toward the target distribution. It has proven that the simplicity of flow matching objectives allows favorable performance in stable training and generation quality compared to solving complex stochastic differential equations in diffusion models. Despite its recent progress in image generation [4], the application of flow matching in robotics domains remains underexplored [5, 6, 7]. We propose the flow matching policy to learn simulated and real-world robot behaviors from raw visual inputs and carry out systematic evaluation.

2 Flow Matching Policy

We build the robot behavioral cloning policy as a generative process of Flow Matching, which constructs a flow vector that continuously transforms a source probability distribution toward a destination distribution. Flow Matching leverages an ordinary differential equation to deterministically mold data distribution, contrasting with diffusion policy which is based on a stochastic differential equation through introducing noise.

2.1 Flow Matching Model

Given a conditional probability density path $p_t(\boldsymbol{x}|\boldsymbol{z})$ and a corresponding conditional vector field $\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{z})$, the objective loss of flow matching could be described as:

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) = \mathbb{E}_{t,q(\boldsymbol{z}),p_t(\boldsymbol{x}|\boldsymbol{z})} \left\| \mathbf{v}_t(\boldsymbol{x}, \boldsymbol{\theta}) - \mathbf{u}_t(\boldsymbol{x}|\boldsymbol{z}) \right\|^2 \tag{1}$$

where $\boldsymbol{x} \sim p_t(\boldsymbol{x}|\boldsymbol{z})$, $t \sim \mathcal{U}[0,1]$ (uniform distribution). Flow matching aims to regress $\mathbf{u}_t(\boldsymbol{x}|\boldsymbol{z})$ with a time-dependent vector field of flow $\mathbf{v}_t(\boldsymbol{x}, \boldsymbol{\theta})$ parameterized as a neural network with weights

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

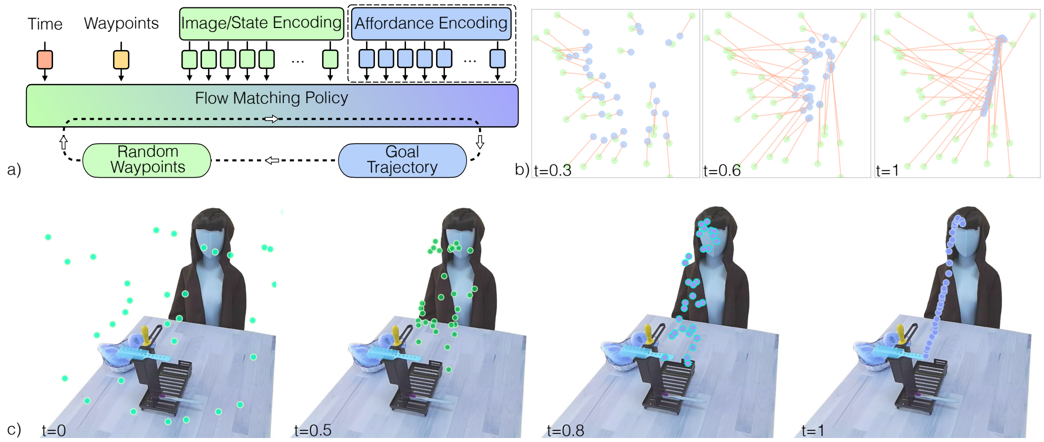

Figure 1: Framework of Flow Matching Policy. (a) At each time step, the policy predicts flow vectors for each waypoint conditioned on visual observation data as well as the waypoints at the current timestep. (b-c) Flow matching transforms random waypoints (green) to the target action trajectory (purple) from timestep 0 to 1. The lines in (b) denote the flow vectors.

$\boldsymbol{\theta}$. $\mathbf{u}_t(\boldsymbol{x}|\boldsymbol{z})$ can be further simplified as:

$$\mathbf{u}_t(\boldsymbol{x}|\boldsymbol{z}) = \boldsymbol{x}_1 - \boldsymbol{x}_0 \qquad \boldsymbol{x}_0 \sim p_0, \boldsymbol{x}_1 \sim p_1$$

p_0 represents a simple base density at time $t = 0$, p_1 denotes the target complicated distribution at time $t = 1$, $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are the corresponding samplings. $\mathbf{v}_t(\boldsymbol{x}, \boldsymbol{\theta})$ could be described as

$$\mathbf{v}_t(\boldsymbol{x}, \boldsymbol{\theta}) = v_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \tag{2}$$

We define $\boldsymbol{x}_t$ as the linear interpolation between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ with respect to time $\boldsymbol{x}_t = t\boldsymbol{x}_1 + (1 - t)\boldsymbol{x}_0$, following the Optimal Transport theory [8]. And $v_{\boldsymbol{\theta}}$ is a network of the flow model. Thus Equation (1) could be reformatted as

$$\mathcal{L}_{\text{FM}}(\boldsymbol{\theta}) = \mathbb{E}_{t,\sim p_0,\sim p_1} \left\| v_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0) \right\|^2 \tag{3}$$

This represents the progression of the scalar flow that transforms data from source to target between time 0 and 1.

Algorithm 1 Robot Flow Matching Policy

Input: observation $\boldsymbol{o}$, target robot actions $\boldsymbol{x}_1$, source random waypoints p_0
Output: flow $\mathbf{v}_{\boldsymbol{\theta}}$
while not converged **do**
  $\boldsymbol{x}_0 \sim p_0$, sample random robot waypoints
  $t \sim \mathcal{U}[0, 1]$, sample time steps
  $\boldsymbol{x}_t = t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$, linear interpolation
  $\mathbf{v}_t(\boldsymbol{x}|\boldsymbol{o}) = v_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t|\boldsymbol{o})$, flow estimation
  $\nabla_{\boldsymbol{\theta}} \left\| v_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t|\boldsymbol{o}) - \dot{\boldsymbol{x}}_t \right\|$, gradient step
end while

Algorithm 2 Robot Diffusion Policy (DDPM)

Input: observation $\boldsymbol{o}$, target robot actions $\boldsymbol{x}_1$, source Gaussian noises p_0
Output: noise $\epsilon_{\boldsymbol{\theta}}$
while not converged **do**
  $\boldsymbol{x}_0 \sim p_0$, sample Gaussian noises
  $t \sim \mathcal{U}[0, 1]$, sample time steps
  $\boldsymbol{x}_t = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, forward process
  $\epsilon_t(\boldsymbol{x}|\boldsymbol{o}) = \epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t|\boldsymbol{o})$, noise estimation
  $\nabla_{\boldsymbol{\theta}} \left\| \epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t|\boldsymbol{o}) - \boldsymbol{\epsilon}_t \right\|$, gradient step
end while

2.2 Flow Matching for Visuomotor Policy Learning

We extend flow matching to learn robot visuomotor policies. This requires two modifications in the formulation: i) modeling the flow estimation conditioned on input observations $\boldsymbol{o}$; ii) changing the output $\boldsymbol{x}$ to represent robot actions. Fig. 1 illustrates our model structures.

Visual observation Conditioning: We modify Equation (2) to allow the model to predict actions conditioned on observations:

$$\mathbf{v}_t(\boldsymbol{x}|\boldsymbol{o}) = v_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t|\boldsymbol{o})$$

	Methods	**2D Trajectory Prediction** (pixel) ↓	**3D Trajectory Prediction** (cm) ↓	**Inference Times** (ms) ↓
Baselines	Diffusion Policy (16-step inference)	0.884	2.096	159.72
	Transformer-based BC	2.797	6.109	7.59
Ours	Flow Matching (Transformer)	2.151	4.842	18.05
	Flow Matching (CNN, 1-step inference)	0.898	2.151	8.53
Ablations	Flow Matching (CNN, 4-step inference)	0.890	2.094	34.99
	Flow Matching (CNN, 8-step inference)	0.887	2.092	70.10
	Flow Matching (CNN, 16-step inference)	**0.841**	**2.091**	100.98

Table 1: The flow matching method achieves the best trajectory estimation accuracy, and faster inference compared to the diffusion policy on Activities of Daily Living dataset.

The visual embeddings $\boldsymbol{o}$ are obtained through ResNet [9]. We also evaluate the Transformer structure in our experiments. Various types of inputs include state-based inputs, RGB images, and visual affordances.

Closed-loop action trajectory prediction: We execute the action trajectory prediction obtained by our flow matching model for a fixed duration before replanning. At each step, the policy takes the observation data $\boldsymbol{o}$ as input and predicts Tp steps of actions, of which Ta steps of actions are executed on the robot without re-planning. Tp is the action prediction horizon and Ta is the action execution horizon. The policy predicts flow vectors $\mathbf{v}_t$ conditioned on visual observation data $\boldsymbol{o}$ with Feature-wise Linear Modulation (FiLM) [10] as well as the interpolated waypoints $\boldsymbol{x}_t$. The flow model $f_{\boldsymbol{\theta}}$ is represented with U-Net [11]. The whole training process of the flow matching policy is illustrated in Algorithm 2.

In our case of robot manipulation, $\boldsymbol{x}_1$ in Equation (3) represents the demonstration robot action trajectories. $\boldsymbol{x}_0$ is the random generated waypoints following a multivariate normal distribution $\boldsymbol{x}_0 \sim \mathcal{N}(0, I)$. $\boldsymbol{x}$ here could denote 6D robot end-effector trajectories or robot joint actions.

Inference: For the inference procedure, random waypoints are sampled from the source distribution and then flowed into the target trajectory by estimating the flow from $t = 0$ to $t = 1$ over steps:

$$\boldsymbol{x}_{t+\Delta t} = x_t + \Delta t f(\boldsymbol{x}_t, t|\boldsymbol{o}), \qquad \text{for } t \in [0, 1] \tag{4}$$

3 Experiments

3.1 Baseline Studies

We compare our flow matching policy against two other robot behavior cloning methods: (i) **Diffusion Policy** [3], and (ii) **Transformer-based** behavior cloning with Mean Square Error Loss, as customary in RVT [12], RT-X [13].

3.2 Datasets and Benchmarks

We benchmark the proposed methods on three datasets: (i) **Push-T** adapted from [14] with RGB images as input, end-effector actions as outputs and $3,000$ epochs of training ; (ii) **Franka Kitchen** proposed in Relay Policy Learning [15] with state-based inputs, robot joint actions as outputs and $4,500$ epochs of training ; (iii) **Real-World Activities of Daily Living Task:** We construct a real-world dataset with 10 tasks across Activities of Daily Living. Each task includes $1,000$ sets of RGB images, demonstrated robot trajectories, and labeled ground truth of affordances. We randomly split the dataset with 80%-20% percentage of training and testing. The results reported here are obtained after $1,000$ epochs of training. The inputs are the RGB image with visual affordance, and the output includes trajectories in both 2D pixel space and 3D Cartesian space.

Methods	Flow Matching (2-step) ↑	Diffusion Policy (2-step) ↑	Flow Matching (16-step) ↑	Diffusion Policy (16-step) ↑	Transformer BC ↑
Push-T	0.8771/0.7111	0.4412/0.1872	**0.9035/0.7490**	0.8840/0.7178	–
Franka Kitchen	0.9750/0.6134	0.2355/0.0527	**0.9960/0.7172**	0.9840/0.6716	–
Activities of Daily Living Task	–	–	**0.82**	0.76	0.44

Table 2: We present the robot performance with different checkpoint selection methods in the format of (max performance) / (average of last checkpoint with 10 trials of replication), with each averaged across 500 different environment initial conditions (5000 in total) for testing. The metric used here is success rate, except for the Push-T task which uses target area coverage.

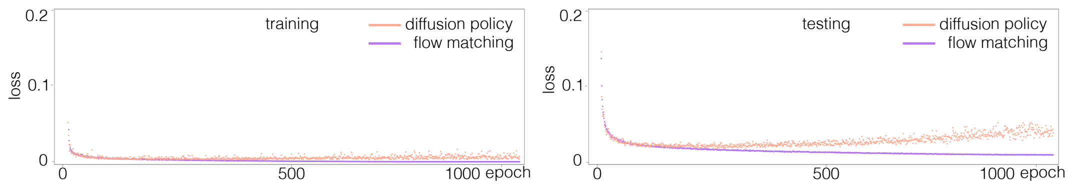

Figure 2: Training and testing loss of flow matching and diffusion policy throughout the training process. Flow matching exhibits greater stability on training and evaluation than the diffusion model.

3.3 Main Results

Table 1 presents the results of flow matching policy on our Activities of Daily Living dataset for robot trajectory learning, comparing against baselines. Table 2 shows the simulated and real robot manipulation evaluation on three benchmarks.

Generation Quality: Table 1 shows that flow matching (CNN-based, 16 steps) outperforms other baselines in terms of 2D and 3D trajectory prediction accuracy. As shown in Table 2, flow matching outperforms other baselines in all three benchmarks of robot manipulation experiments.

Inference Time: In terms of inference time, Table 1 showcases that flow matching with 16 steps achieves faster inference time compared to diffusion policy with 16 steps. We hypothesize that flow matching with linear pointwise flows (following the optimal transport theory) generates straighter flows, and thus causes faster inference. Table 2 showcases better performances of diffusion policy when applying more inference iterations, with a trade-off of longer inference time. Contrarily, flow matching has not shown significant improvements when increasing inference steps. We hypothesize that this is because flow matching trains Continuous Normalizing Flow models, where an ordinary differential equation is solved without learning a series of discrete steps to progressively refine the generated sample. This considerably reduces the inference time for closed-loop robot manipulation, as 1-step flow matching (error: 0.898cm, time: 8.53ms) has achieved comparable performance as 16-step diffusion policy (error: 0.884cm, time: 159.72ms), but prominently lower inference time, as shown in Table 1.

We could also see from Table 1 that CNN-based flow matching achieves better results than transformer-based architecture. We hypothesize that transformer might need additional hyperparameter tuning. Fig. 2 shows the training and testing loss of flow matching and diffusion policy throughout the training process. We can see flow matching exhibits greater stability on both training and evaluation than the diffusion policy.

4 Conclusion

We have systematically studied flow matching framework for supervised robot manipulation, which provides an alternative to diffusion policy. The results suggest forsaking the stochastic construction in favor of more directly learning the probability path, allowing for faster and improved generation. We qualitatively and quantitatively experiment on three simulations and real-world benchmarks to prove the ease of training and evaluation for flow matching.

References

[1] F. Zhang and Y. Demiris. Learning garment manipulation policies toward robot-assisted dressing. *Science robotics*, 7(65):eabm6010, 2022.

[2] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[3] 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.

[4] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[5] M. Braun, N. Jaquier, L. Rozo, and T. Asfour. Riemannian flow matching policy for robot motion learning. *arXiv preprint arXiv:2403.10672*, 2024.

[6] X. Hu, B. Liu, X. Liu, and Q. Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies. *arXiv preprint arXiv:2402.04292*, 2024.

[7] Q. Rouxel, A. Ferrari, S. Ivaldi, and J.-B. Mouret. Flow matching imitation learning for multi-support manipulation. *arXiv preprint arXiv:2407.12381*, 2024.

[8] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[11] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[12] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[13] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[14] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[15] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.