From Potential field to Classifier Guidance Diffusion Policy

Chenghao Lin

***Abstract*— In this work, we enhance diffusion policies, a leading imitation learning method, by integrating obstacle avoidance through classifier guidance. Using the Artificial Potential Field (APF) method, we guide robotic arms away from collisions by reformulating the potential field as an energy-based model. This allows us to compute the probability of collision for a given action and adjust trajectories accordingly. A key advantage of this approach is that it does not require retraining the diffusion policy, making it computationally efficient and easily applicable to pre-trained models. By combining APF with classifier guidance, we hope to achieve robust obstacle avoidance. This method effectively bridges classical control techniques and modern learning-based approaches for better performance in the environments.**

I. INTRODUCTION

From Denoising Diffusion Probabilistic Models[3] (DDPM) to diffusion policy[1][9], diffusion has gained significant attention in recent years and is now being widely applied to robotic tasks. Despite their potential, solving obstacle avoidance problems with diffusion models remains challenging. Many existing works rely on a global perspective, where complete information about all obstacles is assumed. However, this approach is unrealistic for real-world applications.

In this work, we aim to address these limitations by proposing a solution that is more practical and applicable in real-world scenarios. Specifically, we introduce a plug-in module for imitation learning that assists in obstacle avoidance. This plug-in does not require retraining the underlying model; instead, it guides the sampling process using potential fields. We provide mathematical proof to demonstrate the feasibility of our approach.

Our method builds upon and improves ideas from previous studies. For example, [6] using ideas of [8] proposed a guidance approach during sampling by modifying A_{k-1} as follows:

$$A_{k-1} = A_{k-1} - \rho \nabla_{A_k} D(A_0 \mid k, C_{\text{ob}}) \tag{1}$$

where $g_k = \nabla_{A_k} D$ serves as a gradient term for a cost or distance function D and ρ is scale. However, the lack of open-source code and video demonstrations for this work raises questions about its results. [8] uses a guidance term derived from:

$$p(c \mid x_t) = \frac{\exp\{-\lambda E(c, x_t)\}}{Z}, \quad \nabla_{x_t} \log p(c \mid x_t) \propto -\nabla_{x_t} E(c, x_t) \tag{2}$$

However, ignoring the gradient of the normalization factor Z simplifies the equation but is mathematically incorrect. In contrast, our method takes this gradient into account.

Finally, [7] inspired us with its use of the idea of the potential field. However, replacing potential fields with an energy model and learning the energy function gradient using diffusion models appears to be overly complex and misaligned with practical objectives.

In the following sections, we present our mathematical proof to validate the effectiveness and practicality of our proposed method.

II. MATH

A. From Artificial Potential Field to Energy Function

Artificial Potential Field[4] (APF) is a widely used method in robotics for obstacle avoidance and path planning. It defines a potential function $U(A_t, O_t)$, representing the energy landscape shaped by the action trajectory A_t in the action horizon and observation O_t representing obstacles as point cloud from the camera. The robot is guided by the negative gradient of this potential field:

$$F = -\nabla_{A_t} U(A_t, O_t), \tag{3}$$

where F is the virtual force driving the robot toward a goal or away from obstacles. Although APF is effective, it has well-known limitations, such as the risk of getting trapped in local minima and the inability to fully guarantee collision avoidance.

In Energy-Based Models [5](EBMs), this potential function can be naturally interpreted as an energy function:

$$E((A_t, O_t), y) = \begin{cases} U(A_t, O_t), & \text{if } y = 0, \\ C - U(A_t, O_t), & \text{if } y = 1, \end{cases} \tag{4}$$

where C is chosen to ensure the energy function remains bounded and interpretable, which may equal to $\sup(U(O_t))$ or a constant number. We define separate mappings for $y = 0$ (non-collision) and $y = 1$ (collision). The energy function describes how compatible the robotic arm's action A_t and the obstacles observed under O_t with label y. For $y = 0$, higher energy suggests a higher chance of collision, while lower energy indicates safer, more collision-free movements.

B. Energy-Based Model for Probability $p(y \mid A_t, O_t)$

In EBMs, the energy function $E(A_t, O_t, y)$ defines a conditional probability distribution over y:

$$p(y \mid A_t, O_t) = \frac{\exp(-E(A_t, O_t, y))}{Z(A_t, O_t)}, \tag{5}$$

where the partition function $Z(A_t, O_t)$ normalizes the distribution:

$$Z(A_t, O_t) = \sum_{y'} \exp(-E(A_t, O_t, y')). \tag{6}$$

By minimizing energy, the system increases the likelihood of A_t, O_t and y. The conditional probability of $y = 0$ (no collision) can be written as:

$$p(y=0 \mid A_t, O_t) = \frac{\exp(-E(A_t, O_t, 0))}{\exp(-E(A_t, O_t, 0)) + \exp(-E(A_t, O_t, 1))}. \tag{7}$$

Substituting the definitions of $E(A_t, O_t, 0) = U(A_t, O_t)$ and $E(A_t, O_t, 1) = C - U(A_t, O_t)$ from Eq. 4, we obtain:

$$\begin{aligned} p(y=0 \mid A_t, O_t) &= \frac{\exp(-U(A_t, O_t))}{\exp(-U(A_t, O_t)) + \exp(-(C - U(A_t, O_t)))} \\ &= \frac{\exp(-U(A_t, O_t))}{\exp(-U(A_t, O_t)) + \exp(-C + U(A_t, O_t))} \\ &= \frac{1}{1 + \exp(-C + 2U(A_t, O_t))}. \end{aligned} \tag{8}$$

The gradient of $\log p(y = 0 \mid A_t, O_t)$ with respect to A_t is then:

$$\begin{aligned} \nabla_{A_t} \log p(y=0 \mid A_t, O_t) &= \nabla_{A_t} \left(-\log\left(1 + \exp(-C + 2U(A_t, O_t))\right)\right) \\ &= -\frac{\exp(-C + 2U(A_t, O_t))}{1 + \exp(-C + 2U(A_t, O_t))} \\ &\quad \cdot \nabla_{A_t}(-C + 2U(A_t, O_t)) \\ &= -\frac{\exp(-C + 2U(A_t, O_t))}{1 + \exp(-C + 2U(A_t, O_t))} \cdot 2\nabla_{A_t} U(A_t, O_t). \end{aligned} \tag{9}$$

The term $\alpha(U) = \frac{\exp(-C+2U)}{1+\exp(-C+2U)}$ can be interpreted as a weight that varies based on the energy landscape U. Specifically, when U is large, indicating a higher likelihood of collision, $\alpha(U)$ approaches 1. Conversely, when U is small, $\alpha(U)$ approaches 0. The behavior of $\alpha(U)$ is illustrated in Fig. 1. With this definition, Eq. 9 simplifies to:

$$\nabla_{A_t} \log p(y=0 \mid A_t, O_t) = -2\alpha(U) \nabla_{A_t} U(A_t, O_t). \tag{10}$$

Finally, noting that $F = -\nabla_{A_t} U(A_t, O_t)$, we rewrite Eq. 10 as:

$$\nabla_{A_t} \log p(y=0 \mid A_t, O_t) = 2\alpha(U) \cdot F \tag{11}$$

This shows that the gradient of the log-probability for $y = 0$ can be expressed as a scaled version of the force F, where the scaling factor $\alpha(U)$ is a sigmoid-like function depending on the potential energy $U(A_t, O_t)$.

C. Combining Diffusion Policy and Classifier Guidance

The goal is to incorporate y into an existing diffusion-based policy $p(A_t \mid O_t)$ to obtain $p(A_t \mid O_t, y)$. Starting with Bayes' theorem:

$$\begin{aligned} p(A_t \mid O_t, y) &= \frac{p(A_t, O_t, y)}{p(O_t, y)}, & (12) \\ &= \frac{p(A_t, y \mid O_t)}{p(y \mid O_t)}, & (13) \\ &= \frac{p(A_t \mid O_t) \cdot p(y \mid A_t, O_t)}{p(y \mid O_t)}. & (14) \end{aligned}$$

Taking the gradient of the log-probability:

$$\begin{aligned} \nabla_{A_t} \log p(A_t \mid O_t, y) &= \nabla_{A_t} \log\left[p(A_t \mid O_t) \cdot p(y \mid A_t, O_t)\right] \\ &\quad - \nabla_{A_t} \log p(y \mid O_t) \end{aligned} \tag{15}$$

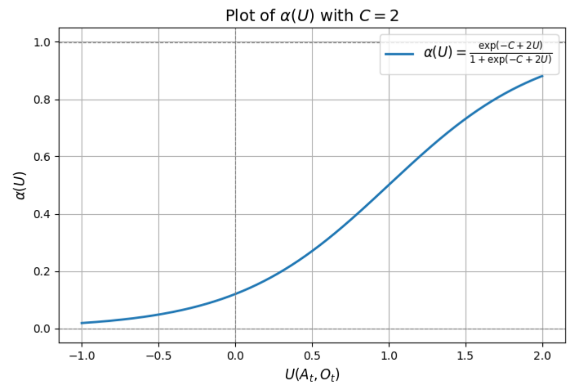

Fig. 1. Plot of $\alpha(U) = \frac{\exp(-C+2U)}{1+\exp(-C+2U)}$ with $C = 2$.

Since $\nabla_{A_t} \log p(y \mid O_t)$ does not depend on A_t, its gradient is zero with respect to A_t. Therefore:

$$\nabla_{A_t} \log p(A_t \mid O_t, y) = \nabla_{A_t} \log p(A_t \mid O_t) + \nabla_{A_t} \log p(y \mid A_t, O_t) \tag{16}$$

Using Eq. 16 and the result from classifier guidance[2], the action sampling update at each step becomes:

$$A_{t-1} \sim \mathcal{N}(\mu + s\Sigma \nabla_{A_t} \log p(y \mid A_t, O_t), \Sigma). \tag{17}$$

If $y = 0$ (non-collision) and based on Eq. 11, the equation can be written as:

$$A_{t-1} \sim \mathcal{N}(\mu + s\Sigma\alpha(U)F, \Sigma). \tag{18}$$

Where s is the gradient scale and μ is from the original diffusion policy output.

III. Lesk and Knowledge-Based Word Sense Disambiguation

Appendix

N/A

Acknowledgment

N/A

References

[1] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[4] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[5] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[6] Hang Li, Qian Feng, Zhi Zheng, Jianxiang Feng, and Alois Knoll. Language-guided object-centric diffusion policy for collision-aware robotic manipulation. *arXiv preprint arXiv:2407.00451*, 2024.

[7] Yunhao Luo, Chen Sun, Joshua B Tenenbaum, and Yilun Du. Potential based diffusion motion planning. *arXiv preprint arXiv:2407.06169*, 2024.

[8] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023.

[9] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.