GenOSIL: Generalized Optimal and Safe Robot Control using Parameter-Conditioned Imitation Learning

Mumuksh Tayal, Manan Tayal and Ravi Prakash

***Abstract*— Ensuring safe and generalizable control remains a fundamental challenge in robotics, particularly when deploying imitation learning in dynamic environments. Traditional behavior cloning (BC) struggles to generalize beyond its training distribution, as it lacks an understanding of the safety-critical reasoning behind expert demonstrations. To address this limitation, we propose GenOSIL, a novel imitation learning framework that explicitly incorporates environment parameters into policy learning via a structured latent representation. Unlike conventional methods that treat the environment as a black box, GenOSIL employs a variational autoencoder (VAE) to encode measurable safety parameters—such as obstacle position, velocity, and geometry—into a latent space that captures intrinsic correlations between expert behavior and environmental constraints. This enables the policy to infer the rationale behind expert trajectories rather than merely replicating them. We validate our approach on two robotic platforms—an autonomous ground vehicle and a Franka Emika Panda manipulator—demonstrating superior safety and goal-reaching performance compared to baseline methods. The simulation and hardware videos can be viewed on the project webpage[1].**

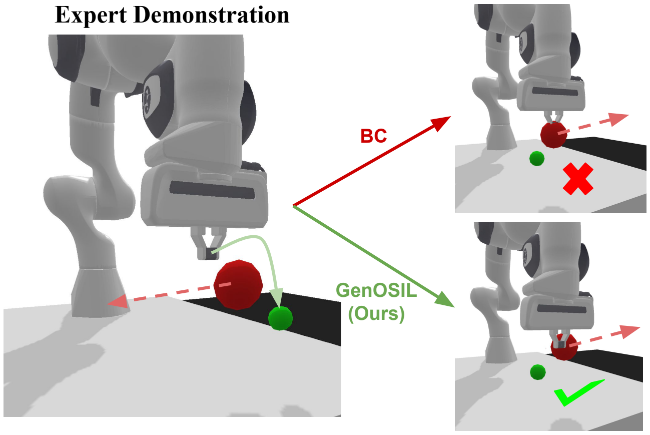

Fig. 1: **Behavior Cloning (BC)** struggles to generalize beyond seen scenarios, often colliding with the dynamic obstacle, while **GenOSIL** learns the underlying objective of avoiding the moving obstacle and successfully navigates dynamic environments

I. Introduction

As autonomous robots become increasingly integrated into real-world applications, ensuring safe and high-performance control remains a fundamental challenge. Imitation learning (IL) has emerged as a promising approach for training control policies by leveraging expert demonstrations, particularly in scenarios where system dynamics are partially known or difficult to model. However, traditional IL methods, such as behavioral cloning (BC), often struggle to generalize beyond the training distribution, leading to performance degradation in deployment, especially in safety-critical scenarios. This challenge is further exacerbated in environments where the robot has only partial observability of its surroundings.

In many practical applications, key safety-related cues (e.g., obstacle position, velocity, and geometry) are directly measurable through onboard sensors. Likewise, task specifications, including goal positions, are often available and may vary across deployments. Despite the availability of these structured environmental cues, conventional IL approaches typically treat the environment as a black box or rely on unsupervised inference of hidden confounders, which can result in suboptimal decision-making and compromised safety.

To address safety constraints in robotic systems, various optimal control strategies have been explored, including Constrained Model Predictive Control (MPC) [1], Hamilton-Jacobi (HJ) reachability-based methods [2], [3], and Control Barrier Functions (CBFs) [4], [5]. While these approaches provide formal safety guarantees, they inherently depend on explicit models of the system and environment dynamics, which are often difficult to obtain in real-world settings. Additionally, these techniques have been integrated with IL to enhance constraint satisfaction in robotic systems. For instance, HJ reachability-based imitation learning [6] explicitly enforces control constraints by computing forward or backward reachable sets to ensure safety. However, such methods are computationally expensive and do not scale well to high-dimensional robotic systems due to the curse of dimensionality. Similarly, CBF-based imitation learning [7] lacks inherent control bounds, often assuming unlimited control authority, which is impractical in real-world applications.

Constrained reinforcement learning (CRL) [8], [9] optimizes policies while enforcing safety constraints, ensuring reliable control in real-world robotic systems. Unlike standard RL, it integrates constraint handling through techniques like Lagrangian relaxation or safety filters to prevent unsafe actions. While effective, these methods often require extensive environment interactions and struggle to generalize across varying safety conditions, limiting their practicality in dynamic, real-world scenarios. Goal-Conditioned Imitation Learning (GCIL) [10], [11] extends traditional imitation

This work was supported in part by PMRF,

All the authors belong to Cyber-Physical Systems, Indian Institute of Science (IISc), Bengaluru. {mumukshtayal, manantayal, ravipr}@iisc.ac.in.

[1] https://mumukshtayal.github.io/GenOSIL/

learning by conditioning policies on goal states, allowing a single policy to achieve multiple objectives. While this improves task adaptability, conventional GCIL methods primarily focus on goal-reaching rather than safety, leading to potential failures in environments with dynamic obstacles or varying safety constraints. These challenges highlight the need for imitation learning frameworks that generalize across diverse environments and safety constraints without requiring explicit system models or computationally expensive reachability analysis.

To address this, we propose GenOSIL, a novel framework that directly incorporates known, measurable safety and task-specific parameters into the policy network. Specifically, GenOSIL, builds upon GCIL by integrating safety-critical environmental factors into a structured latent representation, enabling both task adaptability and safety awareness within a unified framework. To summarize, the key contribution of our paper is as follows:

- **Model-Free Safety Adaptation:** Unlike CBFs and HJ reachability approaches, GenOSIL does not require an explicit system or environment model, making it more applicable to real-world settings where such models are unavailable or inaccurate.
- **Improved Generalization:** By explicitly conditioning the policy on safety parameters and task constraints, GenOSIL generalizes across different environments (i.e., obstacle dynamics), goal locations, and obstacle spawn positions without requiring extensive domain randomization.
- **Scalability to Higher Dimensions:** Unlike reachability-based methods, which suffer from exponential complexity in high-dimensional spaces, GenOSIL scales efficiently by leveraging a learned latent representation of safety and task constraints.
- **Comprehensive Evaluation:** We validate GenOSIL through extensive experiments in simulated robotic environments and hardware implementations. Comparative analyses against baselines demonstrate that GenOSIL achieves superior task performance, enhances safety, and generalizes effectively across varying environmental and task conditions.

Organization: The remainder of this paper is organized as follows: Section II briefs on Safe RL, imitation learning, safe control, and latent variable models. Section III details our proposed GenOSIL framework, including network architectures, loss functions, and training procedures. Experimental results are presented in Section IV. Finally, Section V concludes the paper.

II. Preliminaries

A. Imitation Learning

Imitation learning (IL) is a framework for training policies that map observations to actions using expert demonstrations, in cases where dynamics of the system is unknown or very complex. Various IL approaches exist, including behavioral cloning (BC) [12], which applies supervised learning to replicate expert behavior, DAgger [13], which incrementally expands the training dataset with new expert-labeled state-action pairs, and inverse reinforcement learning (IRL) [14], which derives a cost function that aligns with expert demonstrations by minimizing the inferred cost. While our method is introduced in the context of behavioral cloning, it is not limited to this setting. The theoretical foundations developed in this work apply broadly, ensuring safety guarantees for methods like DAgger and IRL, as they rely on properties of the learned controller rather than the specific IL technique used.

B. Goal-Conditioned Imitation Learning (GCIL)

It is a subdomain of Imitation Learning, where each demonstration is augmented with one or more goal-states that are indicative of the task that the demonstration was provided for. The goal-state contains information that a learning method can leverage to disambiguate demonstrations. Consequently, a goal-conditioned policy, i.e., a policy that includes the goal-state in its condition set, can use a given goal-state to adapt its behavior accordingly. Similarly, goal-states have also extended the domain of reinforcement learning through Goal Conditioned Reinforcement Learning (GCRL), where the agent is not provided expert demonstrations but reward signals instead. Typically these reward signals are difficult to define, especially for complex tasks and environments, providing demonstrations is often a more natural option in such situations. Additionally, the policy rollouts required by GCRL are often expensive in real-world settings. Recent work investigated Goal Conditioned Offline Reinforcement Learning, which does not require these expensive rollouts during training.

C. Variational AutoEncoder (VAE)

Variational AutoEncoders (VAEs) [15] are generative models that learn a probabilistic latent space representation of data. A VAE consists of two neural networks:

- **Encoder** $q_\phi(z|x)$: Maps input data x to a latent variable z by approximating the true posterior distribution.
- **Decoder** $p_\theta(x|z)$: Reconstructs x from z, ensuring meaningful latent representations.

The objective function of a VAE is the Evidence Lower Bound (ELBO):

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)), \quad (1)$$

where the first term maximizes the likelihood of reconstructions, and the second term regularizes the latent space by minimizing the Kullback-Leibler (KL) divergence between the approximate posterior and a prior distribution $p(z)$.

VAEs have been widely adopted for learning structured representations, denoising, and improving generalization in downstream tasks, making them a valuable tool for enhancing imitation learning in RL [16].

Another crucial aspect of training VAE based encoders is the **Reconstruction Loss**, which ensures how well a possible decoder can recreate the original input from the latent representation. This loss term forces the encoder network to capture the salient and most informative details of the input data for downstream tasks.

III. Methodology

In this section, we present GenOSIL, a framework that learns a latent safety embedding to enable robust and generalised safe imitation learning. The method first encodes raw safety parameters (e.g., obstacle position, obstacle velocity, and obstacle radius) into a structured latent variable, and then conditions an imitation policy on both the current state and this latent representation.

A. Problem Formulation

Consider a robotic system with state space $\mathcal{S}$, action space $\mathcal{A}$, and a set of safety parameters $\mathcal{C}$. The safety parameters $c \in \mathcal{C}$ represent critical environmental features such as obstacle positions, velocities, and geometries. The goal is to learn a policy $\pi : \mathcal{S} \times \mathcal{C} \to \mathcal{A}$ that maps states and safety parameters to actions while maintaining safety constraints and accomplishing the desired task. Traditional imitation learning approaches directly map states to actions without explicitly considering safety parameters, leading to poor generalization in safety-critical scenarios. Our approach addresses this limitation by learning a structured latent representation of safety parameters that captures their relationship with expert actions.

B. Latent Safety Embedding

Let $c \in \mathbb{R}^{d_c}$ denote the raw safety parameters. We employ an encoder network $E(\cdot)$ to embed c into a latent variable $z \in \mathbb{R}^{d_z}$. In our VAE-style encoder, we model z using the reparameterization trick:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

where $\mu = f_\mu(c)$ and $\sigma = \exp\left(0.5\, f_{\log var}(c)\right)$ are the outputs of the encoder network, and $\odot$ denotes element-wise multiplication.

We model the latent space using a variational approach to encourage smoothness and interpolation capabilities in the learned representation. This enables the policy to generalize to new safety parameter configurations not seen during training.The encoder architecture consists of fully connected layers with ReLU activations, culminating in parallel output layers for μ and $\log var$.

C. Imitation Policy

The imitation policy $\pi(s, z)$ maps the current state $s \in \mathbb{R}^{d_s}$ and the latent safety variable z to an action $a \in \mathbb{R}^{d_a}$:

$$a = \pi(s, z). \tag{3}$$

Unlike traditional behavior cloning approaches that directly map states to actions, our policy leverages the structured latent representation of safety parameters. This design allows the policy to understand the underlying safety-critical reasoning behind expert demonstrations rather than simply mimicking their actions. The policy network architecture consists of fully connected layers with ReLU activations. The state vector s and latent vector z are concatenated and passed through these layers to produce the action output. This architecture enables the policy to adaptively respond to different safety scenarios while maintaining task performance.

D. Loss Functions

For the training objective of the policy framework, we use a loss function which is composed of these three components as follows:

1) **Imitation Loss:** This loss minimizes the mean squared error (MSE) between the predicted action a and the expert action a^*:

$$\mathcal{L}_{\text{imitation}} = \frac{1}{N} \sum_{i=1}^{N} \|\pi(s_i, z_i) - a_i^*\|^2. \tag{4}$$

2) **KL-Divergence Loss:** This term regularizes the latent space by aligning the encoder's output distribution $q(z|c)$ with a standard normal prior $p(z) \sim \mathcal{N}(0, I)$:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{d_z} \left(1 + \log\left(\sigma_{ij}^2\right) - \mu_{ij}^2 - \sigma_{ij}^2\right). \tag{5}$$

3) **Reconstruction Loss:** To ensure that the latent representation z retains the critical information contained in the raw safety parameters c, we introduce a decoder network $D(\cdot)$ that reconstructs c from z, yielding $\hat{c} = D(z)$. The reconstruction loss is then defined as the mean squared error (MSE) between the original safety parameters and their reconstruction:

$$\mathcal{L}_{\text{Reconstruction}} = \frac{1}{N} \sum_{i=1}^{N} \|c_i - \hat{c}_i\|^2. \tag{6}$$

This loss term forces the encoder to produce a latent embedding that not only supports robust imitation via the policy network but also accurately captures the underlying safety characteristics, thereby promoting an informative and consistent latent space.

The total loss function is given by:

$$\mathcal{L} = \mathcal{L}_{\text{imitation}} + \beta\, \mathcal{L}_{\text{KL}} + \gamma\, \mathcal{L}_{\text{Reconstruction}}, \tag{7}$$

where β & γ are hyperparameters that balance the imitation loss, KL divergence regularization and reconstruction loss. To stabilize training, we implement an adaptive annealing schedule for these hyperparameters, starting with small values and gradually increasing them throughout training.

E. Training Algorithm

Algorithm 1 summarizes the training procedure for GenOSIL. The algorithm integrates the encoder-decoder architecture with the policy network in an end-to-end training framework. The hence trained policy can then be used to sample the actions for incoming states and safety parameters from the franka manipulator.

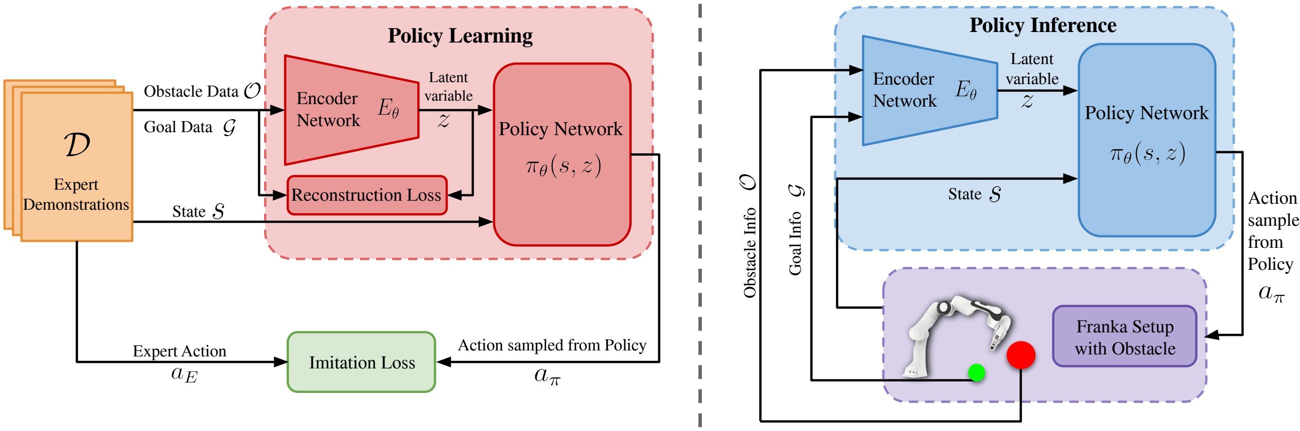
Policy Learning
Obstacle Data $\mathcal{O}$
Goal Data $\mathcal{G}$
$\mathcal{D}$
Expert Demonstrations
State $\mathcal{S}$
Encoder Network E_θ
Latent variable z
Policy Network $\pi_\theta(s, z)$
Reconstruction Loss
Expert Action a_E
Imitation Loss
Action sampled from Policy a_π
Policy Inference
Encoder Network E_θ
Latent variable z
Policy Network $\pi_\theta(s, z)$
State $\mathcal{S}$
Obstacle Info $\mathcal{O}$
Goal Info $\mathcal{G}$
Action sampled from Policy a_π
Franka Setup with Obstacle

Fig. 2: **GenOSIL Architecture:** *(Left)* During training, expert demonstration trajectories—augmented with obstacle parameters, and goal coordinates—are used to learn the Variational Encoder to produce a latent representation (z), which is combined with agent's state to drive training of the Policy Network. *(Right)* At inference, trained architecture deploys the Policy Network on a real agent, generating actions for safe navigation in dynamic environments.

Algorithm 1 Training Procedure for GenOSIL

```
Input: Expert dataset $\mathcal{D} = \{(s_i, c_i, a_i^*)\}_{i=1}^N$, learning rate $\eta$, KL weight $\beta$, reconstruction weight $\gamma$, annealing duration $T_{\text{anneal}}$.
Initialize parameters for encoder $E(\cdot)$, decoder $D(\cdot)$, and policy $\pi(\cdot, \cdot)$.
for each epoch do
    for each mini-batch $\{(s, c, a^*)\}$ in $\mathcal{D}$ do
        Forward Pass:
            Compute latent representation: $z, \mu, \log var \leftarrow E(c)$.
            Predict action: $a \leftarrow \pi(s, z)$.
            Reconstruct safety parameters: $\hat{c} \leftarrow D(z)$.
        Loss Computation:
            Imitation loss: $\mathcal{L}_{\text{imitation}} \leftarrow \|a - a^*\|^2$.
            KL loss: $\mathcal{L}_{\text{KL}} \leftarrow -\frac{1}{2} \sum (1 + \log var - \mu^2 - \exp(\log var))$.
            Reconstruction loss: $\mathcal{L}_{\text{Reconstruction}} \leftarrow \|c - \hat{c}\|^2$.
            Compute current annealing factors: $\beta_t = \beta_{\max} \cdot \min(1, \frac{t}{T_{\text{anneal}}})$, $\gamma_t = \gamma_{\max} \cdot \min(1, \frac{t}{T_{\text{anneal}}})$
            Total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{imitation}} + \beta_t \mathcal{L}_{\text{KL}} + \gamma_t \mathcal{L}_{\text{Reconstruction}}$.
        Backward Pass: Backpropagate $\mathcal{L}$ and update parameters using gradient descent with learning rate $\eta$.
    end for
end for
Output: Trained model parameters.
```

IV. Experiments

We assess the effectiveness of our proposed method, GenOSIL, by benchmarking it against baseline approaches on two simulated tasks –Autonomous Navigation of a Ground Vehicle, and Reach Safe Franka Emika Panda manipulator, as well as its hardware demonstration. Our evaluation focuses on generalization ability, safety performance, and robustness across varies environmental conditions. All experiments are conducted using an AMD Ryzen Threadripper PRO 7975WX (32-core CPU), 32 GB RAM, and an NVIDIA RTX 6000 Ada Generation GPU. Later we also use a Franka Emika Panda for demonstration of hardware implementation of our approach.

A. Baselines and Evaluation Metrics

To evaluate GenOSIL, we compare it against two baselines: Goal-Conditioned Behavior Cloning (GC-BC) [10], which learns policies through behavior cloning under randomized safety conditions without explicitly conditioning on safety parameters, relying on domain randomization for generalization; and Constrained Proximal Policy Optimization (C-PPO) [17], which extends PPO by incorporating safety constraints using Lagrange multipliers to penalize constraint violations during training. GC-BC tests whether exposure to diverse conditions alone enables generalization, while C-PPO provides a reinforcement learning-based comparison that explicitly incorporates constraints. The performance of our method and its baselines are assessed using the following two key metrics:

1) **Safety Rate**: the percentage of test trials in which the agent doesn't collide with the obstacle at any timestamp.
2) **Reach Rate**: the percentage of complete trials where the learned policy successfully reaches the goal. Note that a successfully reached episode is one during which the agent doesn't collide into the obstacle at any point in time during the entire episode. Only such successful reaches are used in calculating the Reach Rate.

B. Autonomous Navigation of a Ground Vehicle

In this task, an agent must reach a parameterized goal while navigating an environment containing a dynamic obstacle. The agent's state is represented as $s = (x, y, \theta)$, where (x, y) denotes position and θ is orientation. The action space consists of linear and angular velocities, $a = (v, \omega)$. The environment features a moving circular obstacle whose position is sampled to ensure a safe margin between

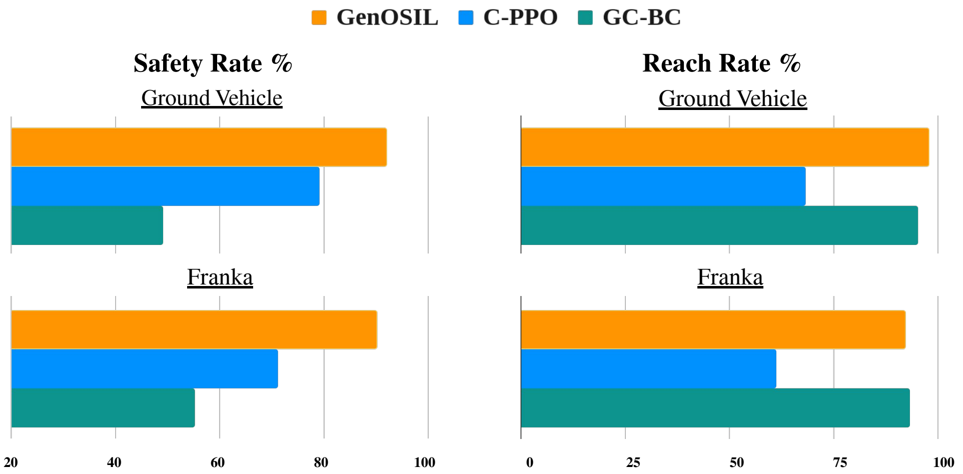

Fig. 3: This figure presents the comparative study between all the methods based on our evaluation matrics. The top plot illustrate the results for the Ground Vehicle Navigation setup, meanwhile, the bottom plot illustrates the Franka Manipulator Task setup. The various bar graphs compare the **Safety Rate (%)** and the **Reach Rate (%)** for both the setups when GenOSIL and the other baseline approaches (GC-BC, C-PPO) are applied.

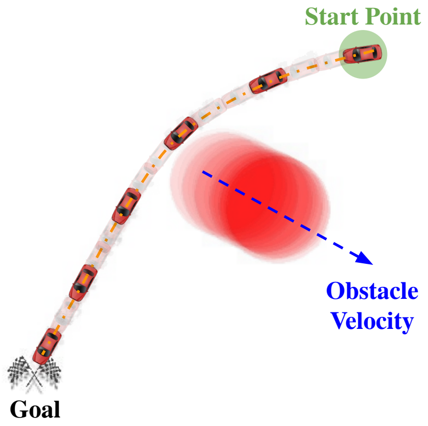

Fig. 4: Autonomous navigation of ground vehicle towards the goal while safely dodging the moving obstacle

the agent's initial state and goal. The obstacle's radius is drawn from a predefined range to introduce variability, and its velocity is dynamically assigned to create unpredictable motion. This setup requires the agent to continuously adapt its trajectory in real time.

Expert Data Generation: Expert demonstrations are generated using a collision cone-based control barrier function (CBF) [18]. The dataset, which includes 10k expert demonstrations, is constructed by randomly sampling initial robot states (s), obstacle parameters ($\mathcal{O}$—position, velocity, and radius), and goal position ($\mathcal{G}$).

Evaluation and Comparative Analysis: We evaluate performance across 1000 sampled test scenarios, ensuring that the expert demonstrations satisfy safety and reach conditions. Figure 3 summarizes the results. GenOSIL outperforms both C-PPO and GC-BC on both the evaluation metrics. It achieves the highest **Safety Rate** while maintaining a superior **Reach Rate**. Although C-PPO effectively enforces safety constraints, it struggles with goal-reaching performance, and GC-BC achieves higher reach rates but suffers from frequent collisions. These results underscore GenOSIL's ability to balance safety and task performance.

C. *Franka Manipulator Task*

In this task, a Franka Panda manipulator must reach a parameterized goal while avoiding obstacles in its workspace. For this experiment, we have used safe-panda-gym simulation environment [19], [20]. The state representation of the manipulator consists of its Cartesian position, $s = (x, y, z)$. The action space comprises end-effector displacement, $a = (dx, dy, dz)$, applied through Position Control. The environment includes static and dynamic obstacles, where obstacle positions, sizes, and velocities are sampled to introduce variability. This setup forces the manipulator to dynamically adjust its trajectory while respecting safety constraints.

Expert Data Generation: Expert demonstrations are collected using the same collision cone-based CBF approach, with datasets generated from randomized initial manipulator states(s), ($\mathcal{O}$—position, velocity, and radius), and goal position ($\mathcal{G}$). This ensures collision-free trajectories for safe reaching.

Evaluation and Comparative Analysis: We evaluate the performance across 1000 sampled test scenarios, ensuring expert demonstrations adhere to safety and reachability conditions. Figure 3 summarizes the results. GenOSIL outperforms both GC-BC and C-PPO in terms of safety and reach rates. While C-PPO achieves commendable safety performance, its goal-reaching capability is compromised, and GC-BC achieves a higher reach rate but at the cost of safety. GenOSIL demonstrates an effective balance between task completion and collision avoidance, validating its application to dynamic robotic manipulation.

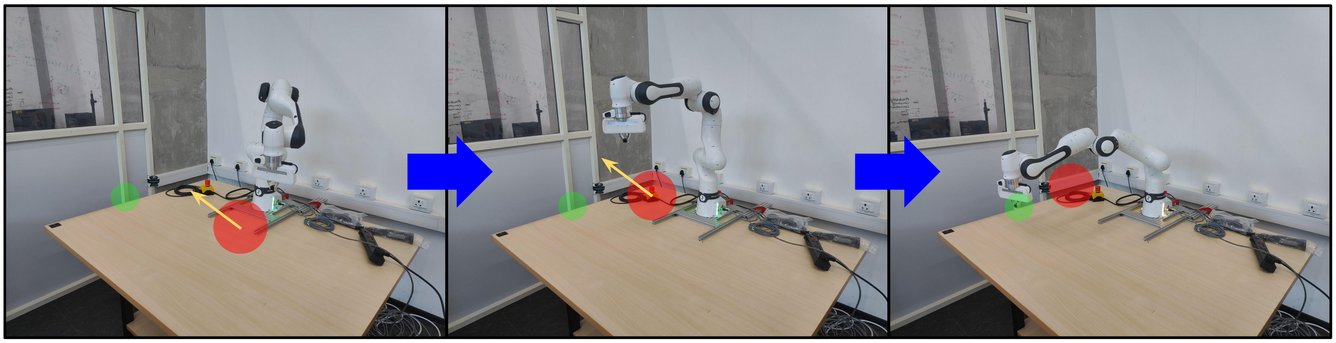

Fig. 5: The figure illustrates the **Franka Panda manipulator** advancing toward its designated target while executing collision avoidance maneuvers in the presence of a dynamic obstacle. The **green region denotes the goal area**, and the **red circle** identifies the **moving obstacle**, with its direction indicated by the arrow.

Hardware Results: We further validate GenOSIL on a physical Franka Emika Panda manipulator. In this setup, virtual obstacles are employed, and environmental parameters (obstacle properties and goal locations) are provided in real time to the policy. Figure 5 presents key frames from the hardware demonstration, confirming successful goal-reaching with effective obstacle avoidance.

V. Conclusion

We presented GenOSIL, a framework for safe and generalizable imitation learning that addresses fundamental limitations in behavior cloning by incorporating environmental safety parameters into policy learning. Experiments on navigation and manipulation tasks demonstrate that GenOSIL outperforms baseline methods in both safety and goal-reaching performance while eliminating the need for explicit system models. Our approach enables effective generalization across varying environmental conditions, making it promising for real-world robotic applications where both performance and safety are critical. Future work will focus on integrating actuation constraints into the framework to further enhance its generalizability and robustness in real-world scenarios.

References

[1] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109899002149

[2] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[3] M. Tayal, A. Singh, S. Kolathaya, and S. Bansal, “A physics-informed machine learning framework for safe and optimal control of autonomous systems,” 2025. [Online]. Available: https://arxiv.org/abs/2502.11057

[4] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[5] M. Tayal, R. Singh, J. Keshavan, and S. Kolathaya, “Control barrier functions in dynamic uavs for kinematic obstacle avoidance: A collision cone approach,” in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 3722–3727.

[6] Y. U. Ciftci, D. Chiu, Z. Feng, G. S. Sukhatme, and S. Bansal, “Safe-gil: Safety guided imitation learning for robotic systems,” *arXiv preprint arXiv:2404.05249*, 2024.

[7] R. K. Cosner, Y. Yue, and A. D. Ames, “End-to-end imitation learning with safety guarantees using control barrier functions,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 5316–5322.

[8] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML’17. JMLR.org, 2017, p. 22–31.

[9] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11797

[10] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, “Goal-conditioned imitation learning,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/c8d3a760ebab631565f8509d84b3b3f1-Paper.pdf

[11] M. Reuss, M. Li, X. Jia, and R. Lioutikov, “Goal-conditioned imitation learning using score-based diffusion policies,” *arXiv preprint arXiv:2304.02532*, 2023.

[12] D. A. Pomerleau, “Efficient training of artificial neural networks for autonomous navigation,” *Neural computation*, vol. 3, no. 1, pp. 88–97, 1991.

[13] S. Ross, G. J. Gordon, and J. A. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” 2011. [Online]. Available: https://arxiv.org/abs/1011.0686

[14] A. Y. Ng and S. J. Russell, “Algorithms for inverse reinforcement learning,” in *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, p. 663–670.

[15] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework,” in *International conference on learning representations*, 2017.

[17] A. Stooke, J. Achiam, and P. Abbeel, “Responsive safety in reinforcement learning by PID lagrangian methods,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9133–9143. [Online]. Available: https://proceedings.mlr.press/v119/stooke20a.html

[18] M. Tayal, B. G. Goswami, K. Rajgopal, R. Singh, T. Rao, J. Keshavan, P. Jagtap, and S. Kolathaya, “A collision cone approach for control barrier functions,” *arXiv preprint arXiv:2403.07043*, 2024.

[19] S. W. Tosin Oseni, “Safe panda gym,” https://github.com/tohsin/Safe-panda-gym, 2022.

[20] Q. Gallouédec, N. Cazin, E. Dellandréa, and L. Chen, “panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning,” *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.