Toward Generalized Application of the Motion Planning Diffusion

Final Project Presentation

Team 6
20243347 JeongWoo Park
20244212 JiHwan Mo

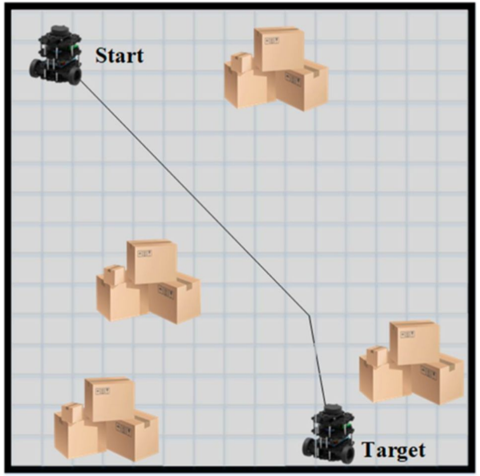
Contents

1. Introduction
2. Related Works
3. Proposed Methodology
4. Experimental Results
5. Discussion
6. Conclusion
7. Team Roles

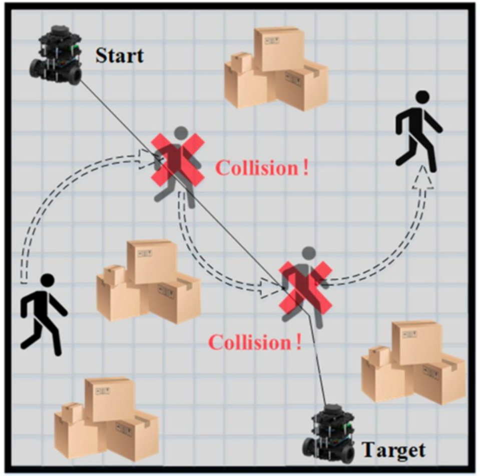
Introduction

Introduction: Problem Statement

- **Challenges**
  - Existing planners : Struggle with **dynamic, real-world complexity**
  - Need : Planners for **general, realistic** environments



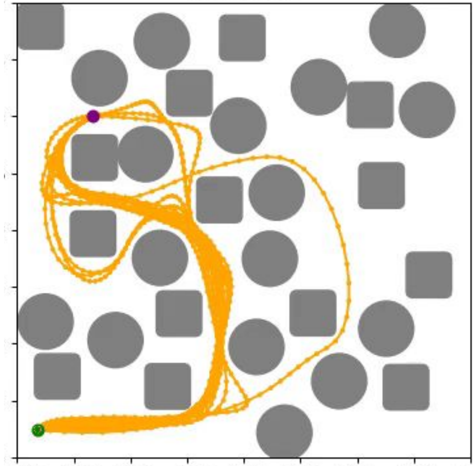
(a) Static Environment



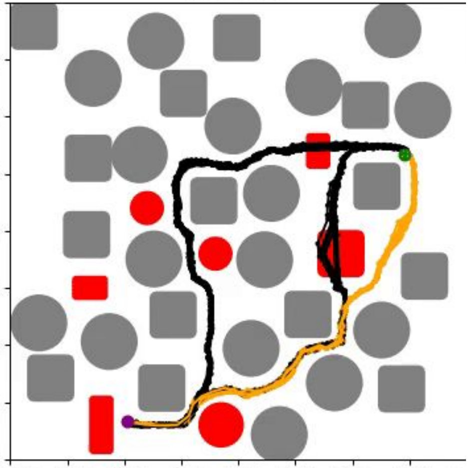
(b) Dynamic Environment

Introduction: MPD

- **Strength:** Multimodal path planning from expert data
- **Limitations:** Requires high-quality data, static environments only
- **Focus:** Analyze dataset impact, extend to dynamic scenes



[Inference without unseen obstacles]

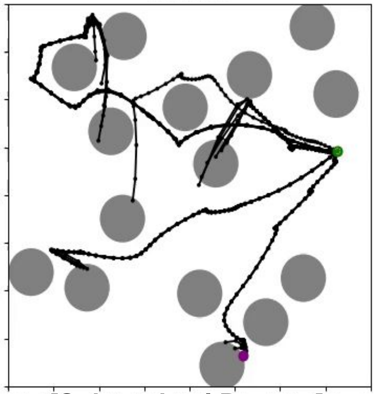


[Inference with unseen obstacles]

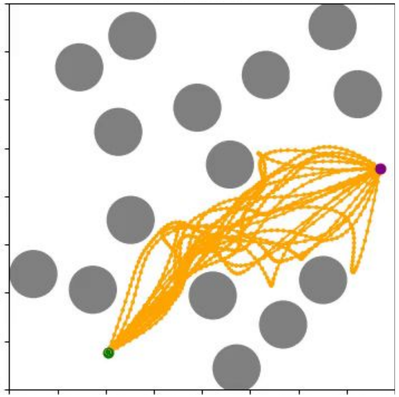
Project Objectives: Generalizing MPD

Overall Goal: Enhance MPD's applicability to real-world scenarios.

- **Objective 1: Analyze Dataset Quality Impact**
  - Compare MPD trained on:
    - Sub-optimal data (less-than-expert)
    - Optimal data (expert demonstrations)
  - Focus: Success rate difference



[Sub-optimal Dataset]



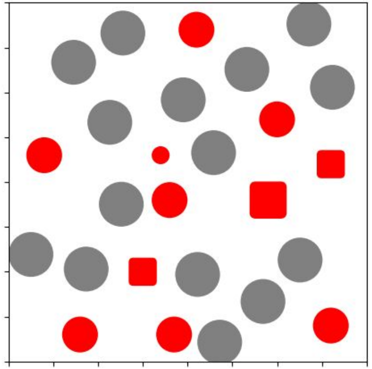
[Optimal Dataset]

Project Objectives: Generalizing MPD

Overall Goal: Enhance MPD's applicability to real-world scenarios.

- **Objective 2: Extend MPD to Dynamic Environments**
  - Evaluate static-trained MPD in dynamic scenes
  - Propose adaptations for dynamic planning
  - Focus: Safety & efficiency with moving obstacles

[Static Environment]



[Dynamic Environment]

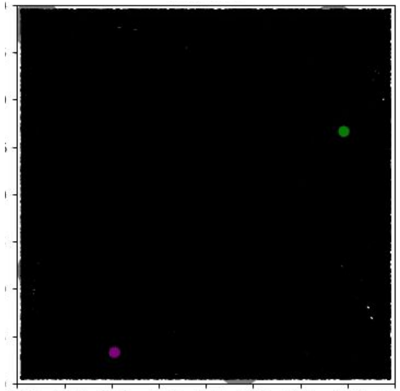
Related Works

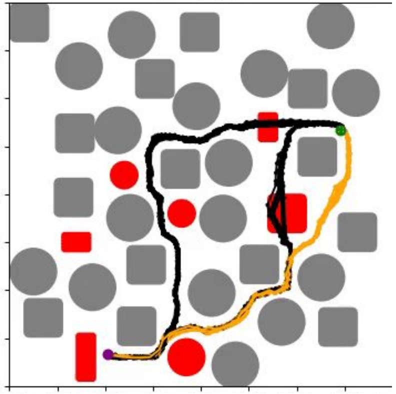
Related Works : MPD vs. Safe RL

<table>
<tr><th></th><th>MPD</th><th>Safe RL</th></tr>
<tr><td>Training Input</td><td>Expert trajectory dataset</td><td>Offline trajectory dataset
(by. a random policy)</td></tr>
<tr><td>Training Goal</td><td>Learn an unconditional trajectory distribution</td><td>Learn trajectory distribution for high reward</td></tr>
<tr><td rowspan="2">Inference Method</td><td colspan="2">Sample trajectories via reverse diffusion</td></tr>
<tr><td>Apply classifier-guided cost gradients at each step</td><td>After each step, project trajectory into feasible constraint set</td></tr>
</table>

Limitations : MPD

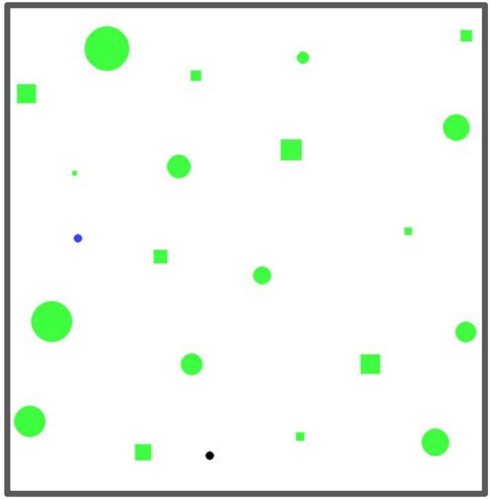
- Assumes **static environments** (no dynamics during train/infer)
- **One-shot trajectory sampling** → can't adapt to changes
- **No dynamic obstacle modeling**
- Dynamic use requires:
  - Constant velocity assumption
  - Known obstacle speeds
  - Manual motion encoding in cost
- **No feedback or re-planning** → unsuitable for online scenarios





Limitations : Safe RL

- Trained on **random policy trajectories**
- In dense scenes:
  - **Early collisions, short failed paths**
- Results:
  - **Few long-horizon successes**
  - Poor generalization in cluttered environments
- **Constraints not learned**, only projected at inference
- **No constraint exposure during training**



Proposed Methodology

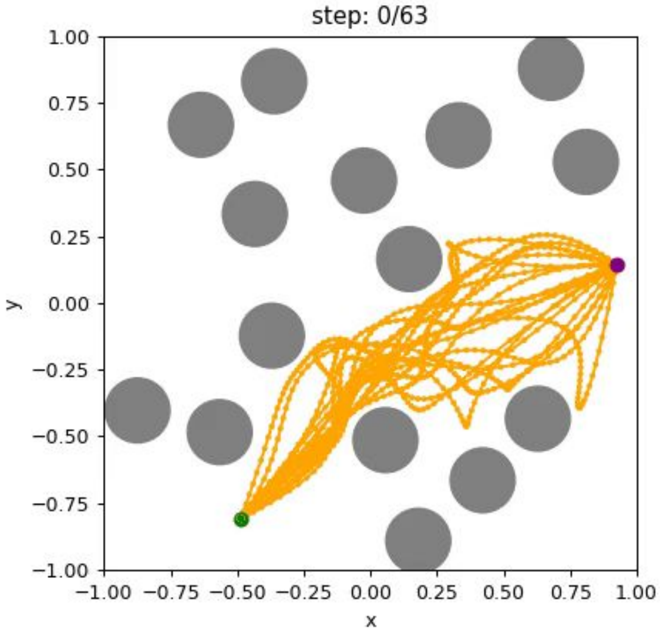
Dataset Quality

Optimal Dataset (Standard MPD)

- Use RRT-Connect + B-spline smoothing
- Produces smooth, collision-free expert paths
- All samples have a fixed length
- Simulates ideal demonstrations

Evaluation & Expected Outcome:

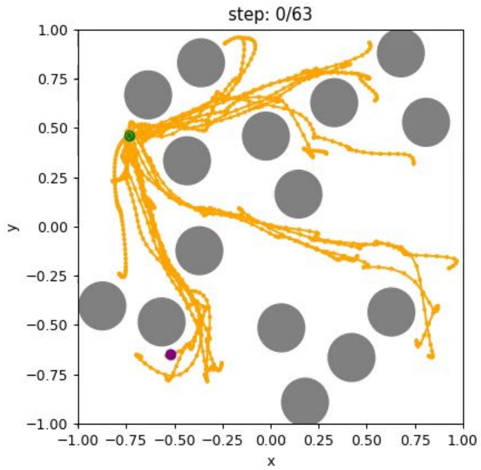
- Train MPD on both dataset types, 50% mixed dataset
- Focus on performance degradation
- Quantifies impact of data quality on planning

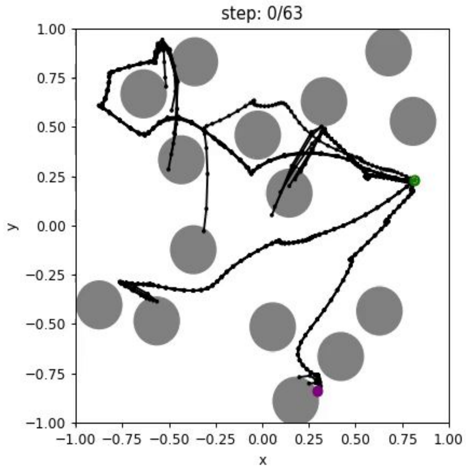


Dataset Quality

Sub-optimal Data (Safe RL-inspired)

- RRT* with zero goal sampling probability, early stopping
  ⇒ Intentionally applied imperfect planning
- Resulting dataset includes a near-misses, collision free paths (up) or paths with **collisions, failures** (down)
- To match the MPD formulation, all samples have a fixed length
- Rationale: Simulates learning from less-than-expert data, common in real-world scenarios or initial stages of learning.



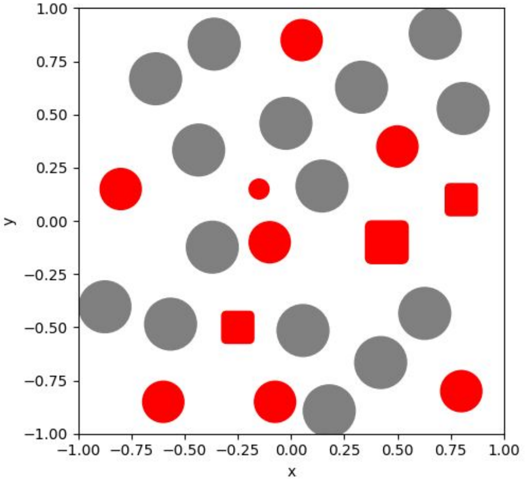


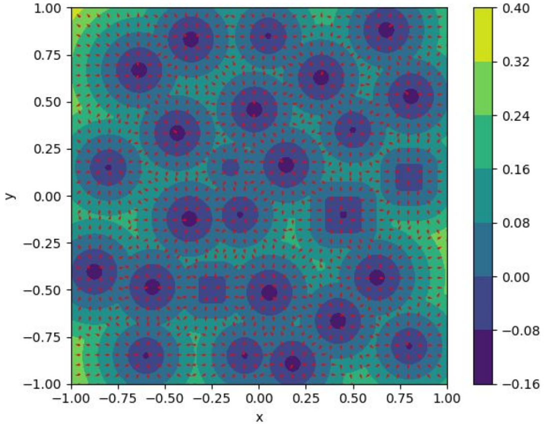
Dynamic Environment

Naive Application (Static Train / Dynamic Test)

- We will only consider PointMass2D-Simple environment from MPD
- Train MPD on **static environments only**
- Add **unseen dynamic obstacles** **only at test time** to avoid unstable learning from dynamic data (e.g., O.O.D, stochasticity)
- Goal: Test MPD's **zero-shot generalization**
- MPD expected to fail on this environment
- Collision cost function depends on time

  ⇒ Harder to compute gradients for guidance



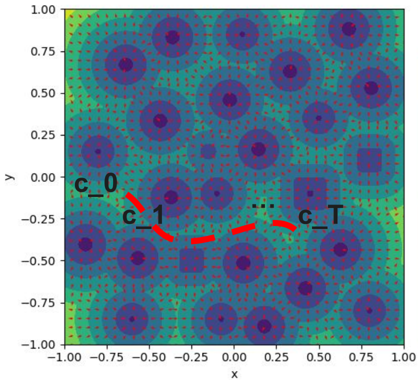


Dynamic Environment

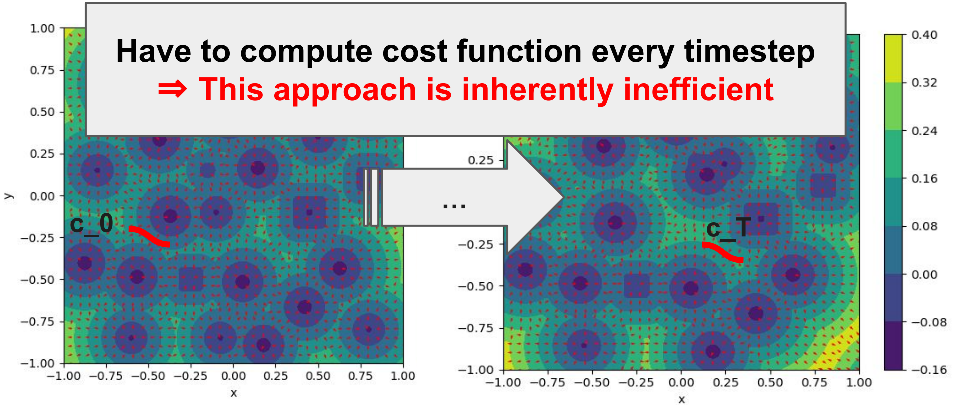
Oracle-Guided Time-Varying Cost

- **Assume full knowledge** of future obstacle motion

  ⇒ Can compute cost function at time **t** during inference of a whole trajectory



[original static guidance]



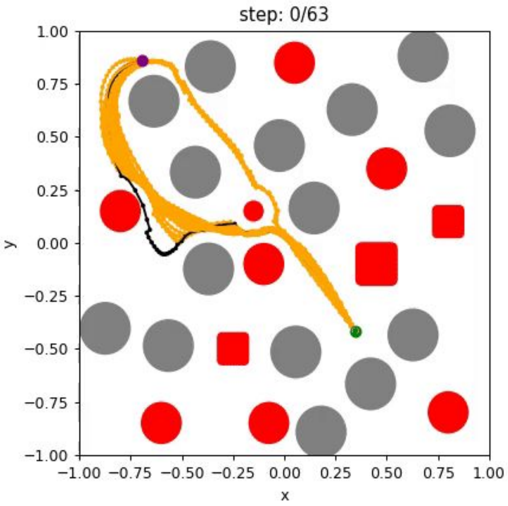
[modified dynamic guidance]

- MPD “**looks ahead**” during trajectory generation (proactive **collision avoidance**)

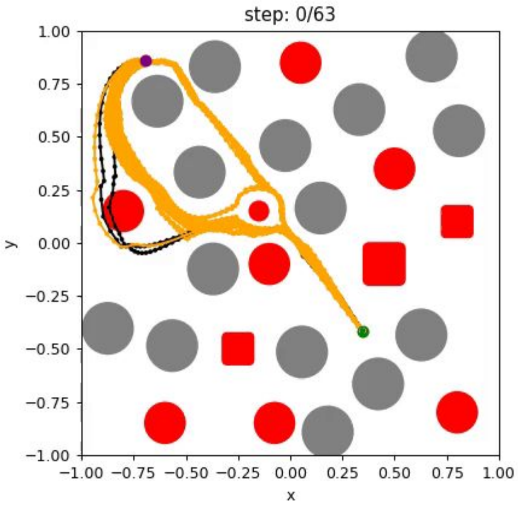
Experimental Results

Result 1. Impact of dataset quality

- PointMass2D-Simple environment



[MPD pre-trained model]

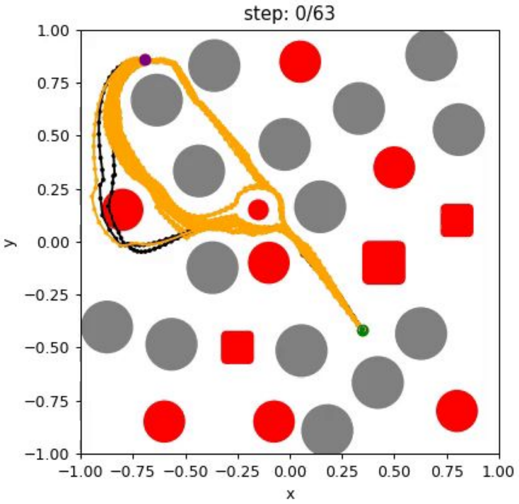


[MPD original dataset]

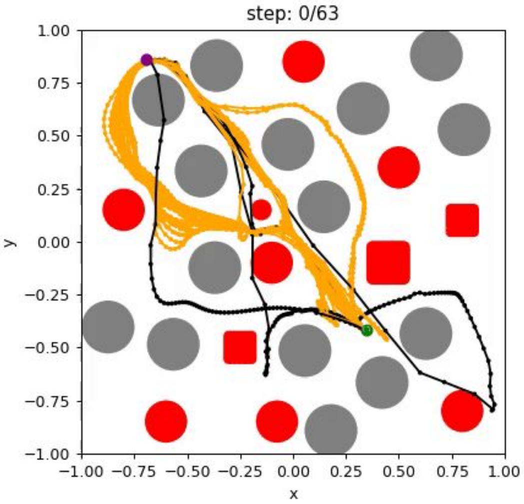
Baseline results are reproduced well

Result 1. Impact of dataset quality

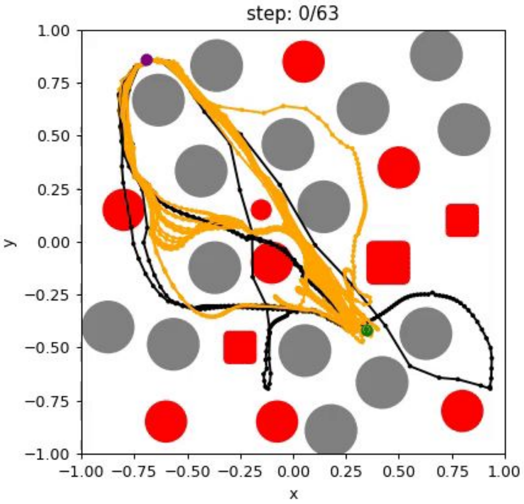
- PointMass2D-Simple environment



[MPD original dataset]



[MPD mixed dataset]



[MPD random dataset]

Using sub-optimal dataset for training led to a performance loss

Result 1. Impact of dataset quality

- Quantitative Comparison

pre-trained : provided weight

Table 1: Impact of training data sub-optimality in Motion Planning Diffusion

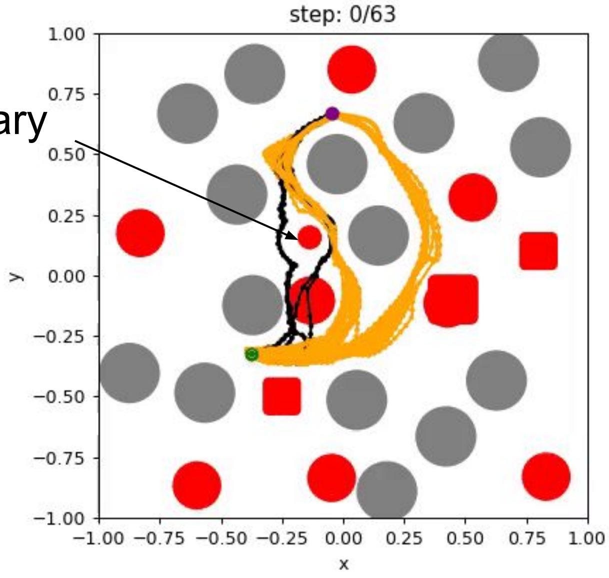
MPD Model	Metrics					
	total t ↓	free traj. ↑	intensity ↓	smoothness ↓	cost path length ↓	cost best ↓	variance ↑
Pre-trained	0.99	98%	0.01	2.97 ± 0.32	2.05 ± 0.08	3.79	0.54
Original data	**1.15**	**96%**	0.17	$\mathbf{3.01 \pm 0.56}$	2.03 ± 0.16	3.53	1.11
Mixed	1.3	92%	**0.12**	3.73 ± 1.26	$\mathbf{1.99 \pm 0.13}$	**2.96**	**3.10**
Random data	1.25	88%	0.27	5.04 ± 1.14	1.97 ± 0.19	4.74	2.37

- Mixed dataset also have some optimal examples, performance may depend on start, goal position, showing stitching capability with diverse paths
- As expected, random data performed worst.

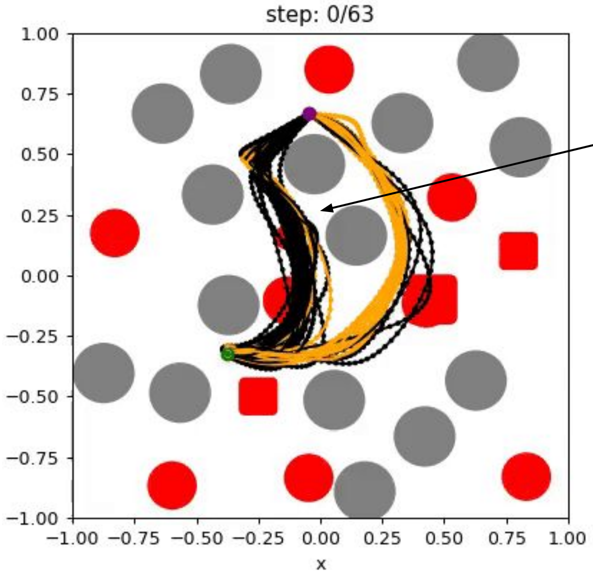
Result 2. Extend MPD to dynamic environments

- Qualitative Comparison 1.

Unnecessary
go-around



[MPD naive application]



[dynamic guidance applied]

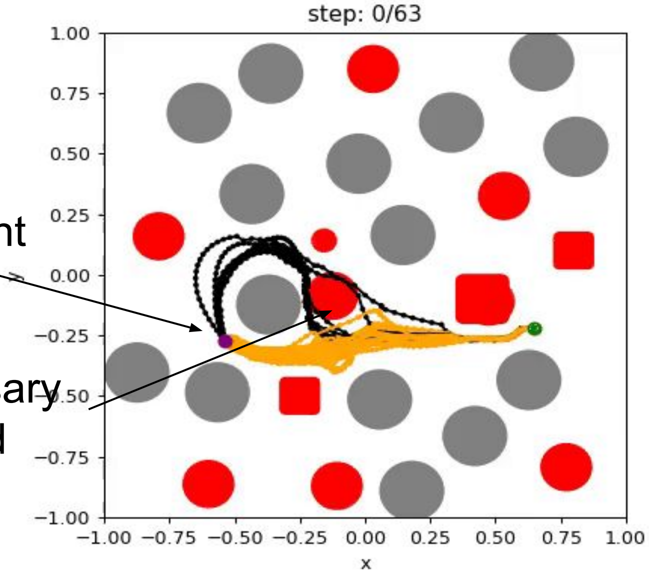
No
go-around

Result 2. Extend MPD to dynamic environments

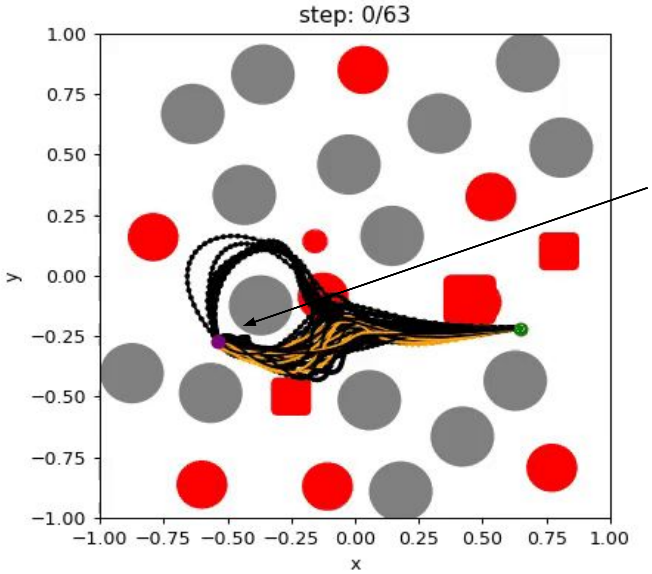
- Qualitative Comparison 2.

Does not consider obstacle movement

Unnecessary go-around



[MPD naive application]



[dynamic guidance applied]

Slow down before collision

Objective 2. Extend MPD to dynamic environments

- Quantitative Comparison with Metrics

```
----------------METRICS----------------
t_total: 1.103 sec
success: 1
percentage free trajs: 92.00
percentage collision intensity: 0.36
cost smoothness: 1.7912, 0.3986
cost path length: 1.3286, 0.1822
cost best: 2.251
variance waypoint: 0.4451
```

[MPD naive application]

```
----------------METRICS----------------
t_total: 21.623 sec
success: 1
percentage free trajs: 86.00
percentage collision intensity: 0.50
cost smoothness: 1.8701, 0.4554
cost path length: 1.3416, 0.1788
cost best: 2.234
variance waypoint: 0.5353
```

[dynamic guidance applied]

Collision intensity : percentage of the waypoints that are in collision

- Minor performance loss in collision free trajectories, smoothness, path length
- Sampling time increased : cost function has to be recomputed every timestep
- Current method : more harm than good

Discussion

Key Observations & Limitations

- **Limited Robot Diversity**
  - Only pointmass tested; complex robots (e.g., Panda) already covered in prior work
- **SDF Instability**
  - Sharp obstacles lead to unstable gradients
  - Smoother SDF representations are needed
- **2D MPD Not Executed**
  - Real-world execution would require inverse kinematics learning
- **Dynamic Data Instability**
  - Training with moving obstacles may cause OOD issues
  - MPD may fail under unseen motion distributions
- **Only Works for Constant-Velocity Obstacles**
  - Not applicable to nonlinear or unpredictable motions

Conclusion & Future Work

Conclusion

- **Dataset Quality**
  - MPD works reasonably with sub-optimal data if state space is well covered
  - Expert data yields better performance in some metrics, however,
    improving robustness using mixed sub-optimal dataset for training seems promising

- **Dynamic Planning**
  - Validated guidance using predicted future obstacle positions (w/o projection)
  - Performance gain was insignificant compared to limitations such as
    Needs perfect information for dynamic obstacles
    Calculating cost function every timestep is computationally heavy

Future Work

- **Dynamic Obstacle in Training**
  - Integrate moving obstacles into training data
- **Visual-Based Perception**
  - Use images to inform MPD about obstacle motion
- **Improved Collision Avoidance**
  - Extend planning with proactive avoidance in dynamic scenes

Thank You