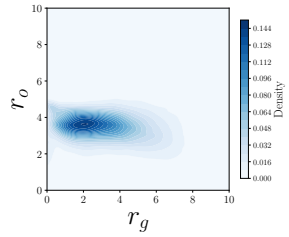
# NaviDiffuser: Tackling Multi-Objective Robot Navigation by Diffusion Model Decision-Making

Xuyang Zhang*, Ziyang Feng*, Quecheng Qiu, Jie Peng, Haoyu Li, and Jianmin Ji†

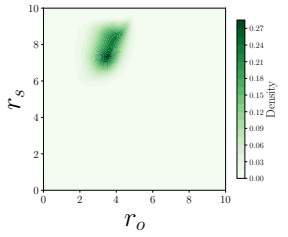
## APPENDIX

### A. Details about the Dataset

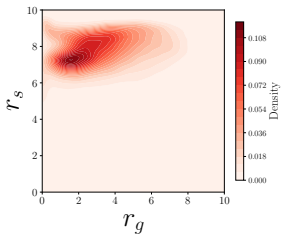
We used the dataset created by PPO policy in our work. However, it is noted that the PPO dataset can be replaced. It is recommend to use the dataset with a wider and more uniform distribution. We used the PPO dataset just because it is enough for training and is easy to build.




(a) $r_g$-$r_o$ plane




(b) $r_o$-$r_s$ plane




(c) $r_s$-$r_g$ plane

Fig. 1: Distribution of reward vector in the PPO dataset.

We conduct a experiment to analysis the diversity of reward vectors in the dataset we used. For each one-second navigation sequence in the dataset, we calculate its average reward vector per step and show these average reward vectors' distribution in Fig. 1. To be honest, the distribution is not uniform. The values of $r_o$ are mainly around 4. Most $r_s$ are located in $[6, 10]$. And the values of $r_g$ are mainly located in $[0, 8]$. However, for any reward vector value, there are some data to cover it with about 0.02 density. These data are created by the totally random data-collecting environment and are all we need to train our approach.

### B. Details about the Network

We provide some hyperparameters of our World and Planning Parts in Tab. I.

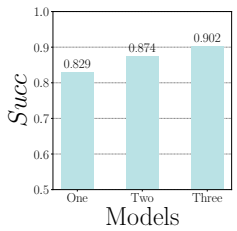
TABLE I: Hyperparameters of World and Planning Parts

| Hyperparameter | World | Planning |
|---|---|---|
| Number of encoder layers | 6 | 6 |
| Number of decoder layers | 8 | 4 |
| Number of attention heads | 12 | 12 |
| Embedding dimension | 768 | 768 |
| Denoising step | 50 | 20 |
| DDIM step | 25 | 10 |
| Beta schedule | cosine | cosine |
| Batch size | 256 | 256 |
| Context length | 8 | 8 |
| Dropout | 0.01 | 0.01 |
| Learning rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Weight decay | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Linear learning rate warmup | $10^4$ | $10^4$ |
| Training steps | $5 \times 10^5$ | $2 \times 10^5$ |

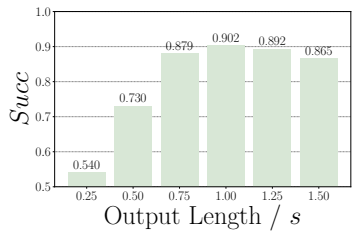
The tensor flow during training is as follows. $s, a, \vec{r}$ in $\tau_H$ and $\tau_F$ (or $\tau_f$) are embedded into tokens through respective MLP layers. The tokens sequence of $\tau_H$ is input into the encoder. The output of the encoder becomes the Key and Value matrix to affect the attention of the decoder. Meanwhile, the tokens sequence of $\tau_F$ is added with standard Gaussian noise $i$ times, concatenated with classification token $c$ and step token $i$, and input into the decoder. The decoder integrates the historical information with the noisy future data to predict the added noise.

In inference, the encoder part is the same. $\tau_F$ is replaced by standard Gaussian noise. And $i$ changes with the number of iterations in the denoising process. Through several denoising iterations described, the decoder generates the corresponding future sequence.

### C. Ablation Study




(a) Different Planning models.




(b) Different output length settings.

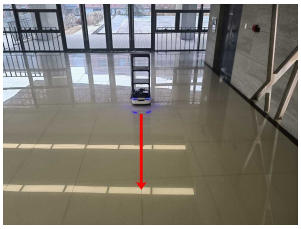
Fig. 2: Arrival rates in the ablation study.

In the ablation study, we first verify whether World Part is beneficial for Planning Part. We train three variants: Model

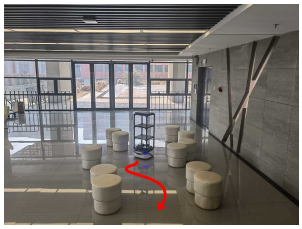
One, built from scratch; Model Two, utilizing shared encoder parameters from World Part; and Model Three, the fine-tuned version of Model Two with augmented data generated by World Part. Their arrival rates in Img-Env's static scenarios are depicted in Fig. 2a. Model Three outperforms the others, followed by Model Two, with Model One trailing. This outcome suggests that leveraging shared network parameters and augmented data from World Part significantly enhances Planning's decision-making skills.

Additionally, we investigate the impact of the output action sequence length of Planning Part. Since Planning is based on Transformer backbone, we can change its output sequence length without retraining. We test Planning with different output lengths in Img-Env's static scenarios. The results are depicted in Fig. 2b. There is an obvious increase in the arrival rate as the output length becomes longer. However, when the output length is longer than $1s$, the arrival rate decreases. The reason can be that the focus on far distance weakens the obstacle-avoiding ability in real time and causes some collisions. We believe that the long-sequence output is beneficial. But the length setting should be selected carefully to get the best performance.

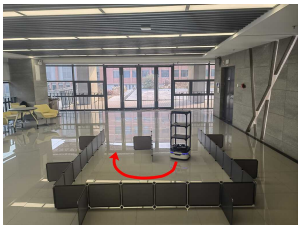
### D. Real World Experiment



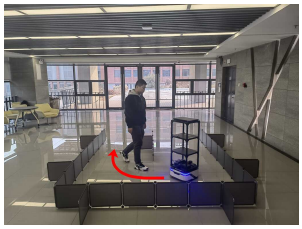
(a) Line



(b) S-shape



(c) U-shape



(d) Dynamic-U

Fig. 3: Real-world testing environment.

For the real-world experiment, we design four representative navigation scenarios. They are shown in Fig. 3. The start and target points in these scenarios are fixed respectively. The pedestrian's trajectories in dynamic scenarios also remain consistent, no matter how the robot moves. We only conduct real-world tests for NaviDiffuser, PPO, and DWA. They perform well in simulation tests. We use TRACER MINI as the real-world robot whose settings are the same as the simulation robot. The laser information comes from an EAI G4 2D Laser and we use AMCL algorithm to obtain the relative position of the target point. For safety considerations, we reduce the robot's maximum linear speed from 1 $m/s$ to 0.6 $m/s$. We test 5 times in each scenario, and the average results are shown in Tab. II.

TABLE II: Performance of our approach and baselines in the real world.

| | Ours | PPO | DWA |
|---|---|---|---|
| $Succ$ | **1.000** | 0.750 | 0.900 |
| $Coll_S$ | **0.000** | 0.100 | 0.100 |
| $Coll_D$ | **0.000** | 0.050 | **0.000** |
| $Stuck$ | **0.000** | 0.100 | **0.000** |
| $v\ m/s$ | 0.479 | 0.333 | **0.581** |
| $w\ rad/s$ | 0.285 | 0.486 | **0.280** |
| $\overline{R}_g$ | 3.775 | 2.032 | **3.884** |
| $\overline{R}_o$ | **5.062** | 5.005 | 3.727 |
| $\overline{R}_s$ | 8.891 | 8.888 | **8.963** |

NaviDiffuser successfully reaches the goal in all tests. DWA collides twice in S-shape. PPO gets stuck in place twice in S-shape and collides three times in Dynamic-U including two wall collisions and one pedestrian collision. With this high arrival rate, NaviDiffuser also provides a high average linear velocity and a low average angular velocity close to DWA. This means that the navigation of NaviDiffuser is fast and smooth. The completion rates of our approach are also competitive. Especially the completion rate of obstacle avoidance is much higher than DWA's. All of these indicate that NaviDiffuser is suitable for deployment in the real world.