NaviFormer: A Data-Driven Robot Navigation Approach via Sequence Modeling and Path Planning with Safety Verification

Xuyang Zhang, Ziyang Feng, Quecheng Qiu, Yu'an Chen, Bei Hua and Jianmin Ji

I. Appendix

Our raw data are sourced from both real-world scenarios and simulators, resulting in the creation of multiple datasets. Each dataset comprises 200,000 $\tau_{\mathbf{D}}$ experience sequences. Within each $\tau_{\mathbf{D}}$, approximately 40 timesteps are captured, with a control interval of 0.25 seconds. The states s_t are derived from 180-degree 10-meter laser data, coupled with the relative position of the goal. As for $p^d_{t:t+l}$, we set $l = 4$, signifying that each future path includes four waypoints. The maximum length of paths is about 1 meter. In terms of the reward function, the constants were set as follows: $\alpha_{\text{collision}} = 1000$, $\beta_{\text{collision}} = 100$, $\alpha_{\text{reach}} = 1000$, $\alpha_{\text{closer}} = 400$, and $\alpha_{\text{further}} = 25$.

The hyperparameters of the sequence modeling network are listed in Tab. I. During the training process, Gaussian noise with a mean of 0 and a variance of 25 is introduced to all $\widehat{R'}_t$ values. The parameter η used for computing the cross-entropy loss of $p^d_{t:t+l}$ is set to 0.5. In the runtime, our implemented return-to-go prediction algorithm and safety verification algorithm are shown in Alg. 1 and Alg. 2, respectively. The safety distance d_{safe} for the safety verification algorithm is configured at 0.2 meters, slightly larger than the robot's radius. The option for re-inference n is set to a maximum of 5 times. NaviFormer is called every 0.25 seconds to generate a safe future path. Subsequently, the dynamic window approach is employed to track this path and effectively control the robot's motion. The real-world robot is equipped with a ThinkPad P15v laptop (Intel Core i7-11800H CPU, Nvidia-T600 4GB GPU), serving as the computing unit. The perception unit consisted of a Hokuyo UTM-30LX Scanning Laser Rangefinder.

TABLE I
Hyperparameters of Sequence Modeling Network

Hyperparameter	**Value**
Number of layers	12
Number of attention heads	12
Embedding dimension	768
Batch size	1024
Context length	8
Nonlinearity	ELU
Dropout	0.01
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Linear learning rate warmup	10^4
Training steps	5×10^4

Algorithm 1: Predict Return-to-go

Input: State s_t, Reward constants $\beta_{\text{collision}}$, α_{reach}, α_{closer}, Number of waypoints in path l, Maximum speed v, Control time interval Δt.
Output: Estimated value of return-to-go $\widehat{R'}_t$.

$\theta = Theta2Goal(s_t)$
$d_{\text{goal}} = Distance2Goal(s_t)$
$d_{\text{obs}} = MinDistance2Obs(s_t)$
$r_{\text{collision}} = r_{\text{reach}} = r_{\text{closer}} = r_{\text{further}} = 0$
if $d_{obs} < 0.5$ **then**
    $r_{\text{collision}} = -\beta_{\text{collision}} \times (0.5 - d_{\text{obs}})^2 \times l$
end
if $d_{goal} < v \times \Delta t \times l$ **then**
    $r_{\text{reach}} = \alpha_{\text{reach}}$
end
$r_{\text{closer}} = Min(0,\ \alpha_{\text{closer}} \times (Cos(\theta) \times v \times \Delta t)^2 \times l)$
return $r_{collision} + r_{reach} + r_{closer} + r_{further}$

Algorithm 2: Safety Verification

Input: Safe distance d_{safe}, Re-inference times n, Sequence modeling network $f_\phi(\tau_{\text{input}}, \text{top}k)$, State s_t, Number of waypoints in path l.
Output: Safe Path $p^d_{t:t+l}$.

$\text{top}k = [0] \times l$
for i **in** $range(n)$ **do**
    $p^d_{t:t+l} = f_\phi(\tau_{\text{input}}, \text{top}k)$
    $distance = [\,]$
    for $waypoint$ **in** $p^d_{t:t+l}$ **do**
        $distance = distance + [MinDistance2Obs(waypoint, s_t),]$
    end
    $distance_{\text{min}} = Min(distance)$
    $index_{\text{min}} = Argmin(distance)$
    if $distance_{min} < d_{safe}$ **then**
        $\text{top}k[index_{\text{min}}] += 1$
    else
        return $p^d_{t:t+l}$
    end
end