Training and Perception for Nomad Navigation

Authors Mathematics and Computing
Indian Institute of Technology

April 8, 2025

Abstract

This report presents our work on implementing the training and perception components of a visual navigation system using diffusion policies, as adapted in the NOMAD framework. We focus on pre-processing, model architecture, and training strategies, leaving out the deployment aspects. We also analyze training performance and evaluate key perception-based metrics.

1 Introduction

Visual navigation in robotics aims to enable agents to traverse environments using visual input. NOMAD (Navigation with Optimal Memory and Action Decoding) is a transformer-based diffusion policy designed for long-horizon, memory-based navigation. Our project involves implementing and training NOMAD while analyzing the perception backbone and its influence on training dynamics.

2 Overview of NOMAD Architecture

The NOMAD architecture comprises three main modules:

- **Perception Backbone**: A ResNet18 feature extractor followed by an attention-based temporal encoder.
- **Trajectory Diffusion Decoder**: A 1D UNet architecture that learns to predict future waypoint trajectories.
- **Action Decoder**: Maps generated waypoints to low-level control commands.

3 Implementation Details

3.1 Environment Setup

3.2 Data Pipeline

We used a pre-collected dataset of trajectories containing RGB observations, actions, and ground-truth waypoints. Data augmentations were not used in our initial experiments.

3.3 Training Procedure

Training was done on a single NVIDIA GPU using a batch size of 64. The training loop involved:

- Calculating diffusion loss from predicted vs ground-truth waypoints.
- Waypoint cosine similarity.
- Auxiliary action prediction losses.

Training checkpoints were saved every epoch. EMA models were also stored.

4 Perception Module

The ResNet18 backbone encodes RGB frames, while a transformer-based encoder maintains temporal context. This allows the policy to act based on history, crucial for long-horizon navigation.

4.1 Cosine Similarity Metrics

We track waypoint cosine similarity to evaluate how well the predicted and ground-truth waypoints align. Early training epochs show increasing cosine similarity, indicating improved waypoint alignment.

5 Results

5.1 Training Metrics

- Final training loss: ~1.11
- Cosine similarity: ~0.47 (multi-action waypoints)
- Distance loss: ~128

5.2 Observations

Loss plateaued after around 5,000 batches. Training logs show improvement in cosine similarity and reduction in loss. Action losses remained stable across UC and GC branches.

6 Challenges and Debugging

7 Conclusion and Future Work

We successfully trained the NOMAD policy and analyzed the perception module. Future work could involve domain randomization, hyperparameter tuning, and evaluating transfer to real-world or simulated environments.

References

1. H. Janner et al., "NOMAD: Planning with Diffusion for Visual Navigation," 2022.
2. Diffusion Policy GitHub Repository: `https://github.com/wayveai/diffusion-policy`