NoMaD: **N**avigati**o**n with Goal-**Ma**sked **D**iffusion

Sehaj Ganjoo, Shobhnik Kriplani,
Abhishek Kumar Jha, Namashivayaa V

IISc Bengaluru
BTech. Mathematics and Computing

April 2025

Motivation and Goal

Robotic learning for navigation in unfamiliar environments requires:

- The ability to perform task-oriented navigation
- Task-agnostic exploration

Issue

Traditionally, these functionalities are tackled by separate systems. Exploration problem can be factorized into:

- **Local Exploration strategies** Objective: Learn control policies that can take diverse, short-horizon actions.
- **Global Exploration strategies** Objective: A high level planner that uses the policies for long-horizon goal-seeking. (Efficient Mapping?)

Solution

Maybe, use a single model to perform both tasks?

What is NoMaD?

NoMaD is a transformer-based diffusion policy designed for long-horizon, memory-based navigation, that is capable of both **goal-conditioned navigation** and **open-ended exploration**.

`NoMaD = EfficientNet + Vision Transformer + Diffusion Policies`

- Can we improve trajectory prediction in robot navigation using denoising diffusion models?
- How can transformer-based memory and temporal context improve performance?

Literature Survey

- **ViNT (Janner et al., 2022):** Vision Transformer for long-horizon visual navigation.
- **Diffusion Policies:** Denoising Diffusion Probabilistic Models (DDPM) for behavior cloning.
- **NOMAD:** Combines ViNT perception with diffusion-based trajectory decoding.

Architecture Overview

- **Perception:** EfficientNet-B0 backbone $\rightarrow$ Transformer-based temporal encoder.
- **Diffusion Decoder:** Conditional UNet1D for generating waypoint sequences.
- **Action Decoder:** Maps waypoints to low-level actions.

Training Procedure

- Dataset: SACSoN / RECON / GoStanford
- Batch size: 32, Epochs: 100
- Optimizer: AdamW, LR: 10^{-4}
- Scheduler: Cosine annealing
- Loss: MSE on predicted noise + temporal distance

$$\mathcal{L}_{NoMaD} = MSE(\epsilon, \hat{\epsilon}) + \lambda \cdot MSE(d(o_t, o_g), f_d(c_t))$$

Experiments and Results

Metrics:

- Diffusion Loss ≈ 1.11
- Distance Loss ≈ 128
- Cosine Similarity ≈ 0.47

Comparison with ViNT:

- Similar performance in goal-conditioned tasks
- No performance degradation when adding diffusion

Challenges Faced

- CUDA Out Of Memory errors on limited GPU
- Module import issues with nested folder structures
- Gradients not propagating due to detached variables

Team Contributions

Sehaj Ganjoo:

- Set up training pipeline and environment
- Integrated and debugged diffusion model
- Wrote training script and logging tools
- Conducted experiments and generated plots
- Created report and presentation

Conclusion and Future Work

- Successfully trained NOMAD using diffusion for visual navigation
- Showed compatibility with ViNT-based perception
- Future work:
  - Evaluate in simulation / real-world
  - Improve runtime performance
  - Try larger ViTs and alternate decoders

Q&A

Thank you!
Questions are welcome.