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 navigation in unfamiliar environments requires:

- Task-oriented navigation — reaching specified goals
- Task-agnostic exploration — discovering and mapping new areas

The Challenge

These two objectives are typically handled by *separate systems*.

Exploration can be decomposed into:

- **Local Exploration:** Learning short-horizon control policies for diverse actions
- **Global Planning:** Using those policies to achieve long-horizon, goal-directed behavior

Key Question

Can a *single model* unify both tasks — exploration and navigation?

What is NoMaD?

NoMaD is a transformer-based diffusion policy designed for long-horizon, memory-efficient navigation.

It supports both:

- **Goal-conditioned navigation** — moving towards a specified visual goal
- **Open-ended exploration** — learning diverse behaviors without explicit goals

$$\texttt{NoMaD} = \{\texttt{EfficientNet + Vision Transformer}\} \leftarrow \texttt{ViNT} + \texttt{Diffusion Policies}$$

It combines a transformer backbone to encode the high-dimensional visual stream, with diffusion models that predict a sequence of future actions in a generative manner.

Overview of NoMaD Architecture

Visual Goal-Conditioned Navigation

Backbone: ViNT (Visual Navigation Transformer)
How does ViNT work?

- Recieves: A sequence of past and current observations $o_t = o_{t-P:t}$
- **Visual Encoder:** Each observation is processed using an EfficientNet-B0 encoder to extract feature embeddings.

Overview of NoMaD Architecture

Visual Goal-Conditioned Navigation

Backbone: ViNT (Visual Navigation Transformer)
How does ViNT work?

- Recieves: A sequence of past and current observations $o_t = o_{t-P:t}$
- **Visual Encoder:** Each observation is processed using an EfficientNet-B0 encoder to extract feature embeddings.

`EfficientNet?`

- A new method of Scaling CNNs to improve accuracy and efficiency
- It uses a compound scaling method to uniformly scale all dimensions of depth, width, and resolution.

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.