Setup of a Framework for Employing Foundation Models in Autonomous Driving

Md Yamin Mollah, Md Shiful Islam, Md Fuad Hasan, Elvis Soans, Manav Joshi
Technische Hochschule Ingolstadt

***Abstract*—Machine learning has been revolutionized by foundation models, which allow practitioners to use smaller datasets than those needed to train models from scratch to produce generalizable answers for particular tasks. Under inadequate oversight, these models are usually pre-trained on sizable, varied datasets, utilizing a far greater amount of data than is accessible for separate downstream applications.**

Here, we present a framework for foundation models created especially for vision-based autonomous driving using the Visual Navigation Transformer (ViNT). Navigational affordances are integrated into the model, allowing for effective adaptability to a range of downstream activities like autonomous driving.

Key features of ViNT include:

Diverse Training: Hundreds of hours of navigation data gathered from multiple robotic platforms have been used to pre-train ViNT on a large number of existing robotic navigation datasets. ViNT can beat specialized models trained on smaller datasets thanks to this training strategy, which promotes positive transfer.

Exploration and Adaptability: ViNT can successfully explore new settings when it is outfitted with diffusion-based goal ideas. When combined with long-range heuristics, it also resolves navigation issues at the kilometer scale.

Work Flexibility: ViNT can dynamically adjust to new work requirements. It uses a prompt-tuning-inspired method, for example, in which the encodings of other modalities (such as GPS waypoints or turn-by-turn directions) can be used in place of its goal encoder and embedded into the same goal token space.

I. Introduction

Recent developments in machine learning have shown impressive results in fields such as visual perception, natural language processing, and others. These accomplishments are mostly ascribed to "foundation models," which are pre-trained on huge datasets and may be fine-tuned, prompt-tuned, or zero-shot transferred to new tasks. However, because of the variety of locations, platforms, environments and applications involved, adopting this paradigm to autonomous driving is difficult. This project shows a framework for foundation models using ViNT architecture and it's pre-trained checkpoints.

II. Overview of ViNT Foundation Model

A pre-trained system is referred to as a foundation model if it can:

Use zero-shot in new contexts, taking into account differences in sensors, robotic bodies, and surroundings. Adjust to downstream tasks, which may involve a variety of goals, actions, and objectives. This work focuses on visual navigation, in which robots only use egocentric visual inputs to navigate. For this area, a general pre-trained model should be able to support a variety of navigation tasks, generalize across different platforms and surroundings, and allow for fine-tuning with less data. A strong baseline policy for navigation would be provided by such a model, allowing applications to be customized for particular robotic platforms and domains.

The Visual Navigation Transformer (ViNT), a cross-embodiment foundation model for visual navigation with robust zero-shot generalization, is presented in this study to meet these needs. Among ViNT's primary characteristics are:

- General Training Objective: Using information from any mobile robot platform, ViNT is trained to accomplish objectives given by camera images.
- Exploration Capability: ViNT can navigate and map previously unexplored environments thanks to a revolutionary exploration algorithm that uses a diffusion model to suggest short-horizon goals.
- Versatility: ViNT shows that it can travel kilometer-scale outdoor areas without assistance, manage new robots zero-shot, and map indoor spaces.
- Adaptability: New task modalities, such GPS waypoints or high-level routing commands, can be accommodated by fine-tuning the model with little input.
- Emergent Behaviors: Qualitative analysis identifies emergent behaviors, such as dynamic pedestrian navigation and implicit preferences.

With its versatility across different robots and activities, ViNT is a major step towards general-purpose foundation models for robotics. ViNT lays the groundwork for a variety of mobile robotic applications by offering a strong and adaptable framework, opening the door for additional developments in robotic navigation.

Fig: Overview of ViNT Foundation Model

III. Related Work

Research has focused on using big, diversified robotic datasets for a variety of applications, allowing data exchange across similar robotic platforms to create models that are more

broadly applicable. However, differences in dynamics and camera setups (e.g., focal length, field of view, and extrinsics) provide difficulties in mobile robotics such as autonomous driving. Current methods frequently depend on:

- Small Real-World Datasets: These are not generalizable and are restricted to a specific robotic platform.
- Simulation-Based Training: This transfers learn policies by using coupled robot and environment models.

However, this study takes a data-driven method, using a variety of real-world datasets gathered from various robotic and driving environments to learn navigation patterns. Creating a foundation model that can adapt to different downstream jobs, either zero-shot or with little fine-tuning, is the main goal.

Training a visual navigation policy that can handle a variety of downstream jobs, including,

- navigating toward GPS targets.
- achieving the desired pictures.
- doing driving that is skill-conditioned.

In order to expand on well-known visual navigation strategies, this work uses:

- Topological Graphs: For representing environments spatially.
- Learned Policies: For robot control at a low level.
- Learned Heuristics:To steer robots in unfamiliar surroundings, heuristics were learned.

This work differs from previous research in that it focuses on training a single generalist model instead of specific solutions for every task. Demonstrating that a high-capacity model can be modified for a variety of autonomous applications with little extra data is the goal.

Relevant Literature and ViNT's Innovative Input Works like RT-1, I2O, and GNM, which investigate generality across settings and embodiments, serve as inspiration for ViNT. Nonetheless, ViNT differs in a number of ways:

- RT-1: Concentrates on obeying a variety of directions. On the other hand, ViNT places emphasis on fine-tuning a single model to fit a variety of automated platforms and tasks.
- I2O: I2O is excellent at translating policies from simulation to real-world situations, ViNT focuses on developing strong navigation policies that can be applied to a variety of downstream tasks without relying on the underlying algorithm.
- GNM: Uses diverse RGB datasets to train policies for image-goal navigation with zero shots. By focusing on adaptation across tasks and embodiments while maintaining good zero-shot performance, ViNT goes beyond this.

ViNT's Fundamental Advantages

For visual navigation, ViNT trains a single generalist policy that is particularly good at:

- Task Diversity: Solving multiple downstream navigation tasks.
- Embodiment Adaptability: Adapting to different robotic platforms.
- Zero-Shot Deployment: Navigating effectively in new environments without prior fine-tuning.

Effectively navigating in unfamiliar situations without any prior fine-tuning is known as "zero-shot deployment." ViNT positions itself as a flexible mobile robotics solution by emphasizing generalization and adaptation, which enables it to manage a variety of tasks and manifestations with little more training.

IV. Architecture of the ViNT Model

Robots can navigate towards a subgoal that is described by an image observation according to ViNT's image-goal navigation concept. This method is appropriate for a variety of datasets and broad generalization since it makes few assumptions and uses video and action data instead of ground-truth localization, semantic labels, or metadata.

ViNT predicts (i) the time steps to reach the subgoal (dynamical distance) and (ii) a series of future actions leading to the goal by analyzing past and present visual observations as well as a subgoal image. Based on a Transformer architecture, the 31M-parameter model may be adjusted for tasks farther down the line and is tailored for effective inference on robots with limited resources. Current and historical observations are encoded into feature vectors with the use of EfficientNet-B0 to tokenize inputs. ViNT uses a goal fusion encoder, which stacks the pictures of the current and goal observations and processes them together, to overcome the low performance of solo goal encoding. These tokens are fed into a 4-layer decoder-only Transformer with multi-head attention, along with positional encoding. Strong and adaptable image-based navigation is made possible by this design.Using a minibatch of trajectories, ViNT is trained. The temporal context is formed by P consecutive observations, and a future observation is chosen at random from a predetermined range to act as the subgoal. ViNT uses an embodiment-agnostic action space based on relative waypoints, normalized by the robot's top speed, to facilitate training across robots with different sizes, speeds, and dynamics. A robot-specific controller un-normalizes and tracks these waypoints during deployment, whereas this abstraction guarantees consistency between platforms.Over 100 hours of real-world navigation trajectories from eight different robotic platforms are used in the training process. This dataset allows ViNT to generalize well because it encompasses a large range of surroundings, dynamics, and camera setups. After training, ViNT may be used on any robot that has a low-level velocity tracking controller and an onboard camera, making it incredibly adaptable for real-world scenarios.

V. Methodology

This framework of foundation models for autonomous driving uses ViNT and NoMaD pre-trained models (Checkpoints). NoMaD uses goal masking diffusion policies navigation and exploration which is an advanced version of ViNT model. Also NoMaD is developed on top of ViNT architecture. Here is the

step-by-step procedure we followed to evaluate our real-world datasets for autonomous driving using both ViNT and NoMaD pre-trained checkpoints:

A. Pre-requisites

The codebase requires a workstation with Python 3.7+, Ubuntu (tested on 18.04 and 20.04), and a GPU with CUDA 10+ installed. Although you can adapt it to operate with different virtual environment packages or a native configuration, it also presumes access to Conda.

B. Project Setup

Run the following commands inside the root directory of the project.In our case the default name of the root directory is `foundation_model`

1) Set up the conda environment:
   ```
   conda env create -f train
   /train_environment.yaml
   ```
2) Activate the conda environment with the exact name defined in `train_environment.yaml`:
   ```
   conda activate vint_train
   ```
3) Install the `vint_train` packages:
   ```
   pip install -e train/
   ```
4) Install the `diffusion_policy` package from the following repo:
   ```
   https://github.com/real-stanford
   /diffusion_policy
   ```
   Commands:
   ```
   git clone git@github
   .com:real-stanford/diffusion_policy.git

   pip install -e diffusion_policy/
   ```

C. Data Processing

The datsets should be processed as the following structure:

```
<dataset_name>
+-- <name_of_traj1>
|   +-- 0.jpg
|   +-- 1.jpg
|   +-- ...
|   +-- T_1.jpg
|   \-- traj_data.pkl
+-- <name_of_traj2>
|   +-- 0.jpg
|   +-- 1.jpg
|   +-- ...
|   +-- T_2.jpg
|   \-- traj_data.pkl
...
\-- <name_of_trajN>
    +-- 0.jpg
    +-- 1.jpg
    +-- ...
    +-- T_N.jpg
    \-- traj_data.pkl
```

1) Add the data set root directory path to the relevant model's config files i.e. vint.yaml for ViNT and nomad.yaml for NoMaD models.
2) Run the command inside the root directory of the project:
   ```
   python <path to the data_split.py file>
   ```
   After running this command, the processed data-split should the following structure inside `vint_release/train/vint_train /data/data_splits/`

   ```
   <dataset_name>
   +-- train
   |   \-- traj_names.txt
   \-- test
       \-- traj_names.txt
   ```

   So now the data is structured for both training and evaluation.

D. Training with Pre-trained Checkpoints

1) For training set train: False inside the related model config file: For example to train with ViNT checkpoints, make this change to `/foundation_model/train/config/vint.yaml`
2) Download the pre-trained models from this link: Pre-trained models
3) Add:`load_run:<project_name>/ <log_run_name>` to your `.yaml` config file in `foundation_model/train/config/`. The *.pth of the file you are loading to be saved in this file structure and renamed to "latest": `foundation_model/train /logs/<project_name>/<log_run_name> /latest.pth`. This makes it easy to train from the checkpoint of a previous run since logs are saved this way by default. Note: if you are loading a checkpoint from a previous run, check for the name the run in the `foundation_model/train/logs/ <project_name>/`, since the code appends a string of the date to each `run_name` specified in the config yaml file of the run to avoid duplicate run names.
4) Run this command inside the `foundation_model/train/` directory: `python train.py -c <path_of_train_config_file>`

E. Data Evaluation with Pre-trained Checkpoints

The same approach as the previous training should be followed for evaluation except the "train" property in the config file should be set to "False" for only evaluation. Otherwise both training and evaluation happens sequentially.

VI. Evaluation Results

Here are some output of data evaluation with pre-trained ViNT and NoMaD checkpoints

A. THI Test Action Prediction

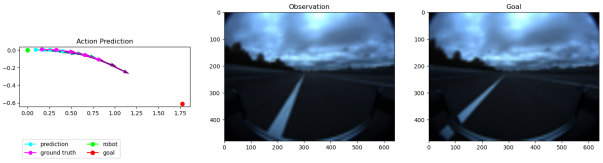
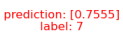


Fig: THI Data Test Action Prediction



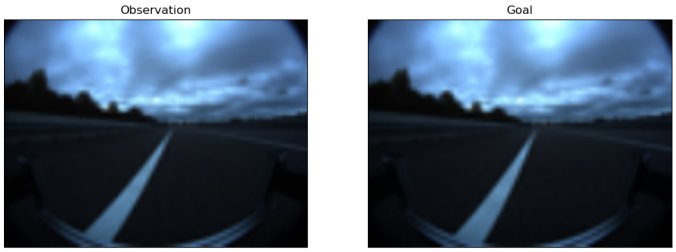


Fig: THI Data Test Distance Prediction

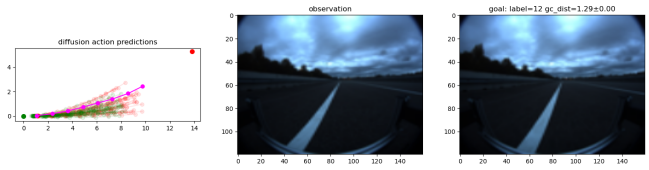


Fig: THI Data Test Action Samples

VII. Comparison

Metric	ViNT Pre-trained Checkpoints	Single Data Set Trained Model
Action Loss	`gc_action_loss: 4.4738`, `uc_action_loss: 4.3666`	`action_loss: 0.85798`
Distance Loss	`gc_dist_loss: 131.4048`	`dist_loss: 127.74493`
Cosine Similarity (Waypoints)	`gc: 0.92013`, `uc: 0.91514`, `gc_multi: 0.85984`, `uc_multi: 0.85367`	`0.97082`, `multi_action_waypts_cos_sim: 0.95982`

Fig: COMPARISON OF VINT PRE-TRAINED CHECKPOINTS AND SINGLE DATA SET TRAINED MODEL

VIII. Conclusion

Systems that enable reliable zero-shot navigation and effective adaptation to a variety of activities and autonomous platforms are advanced by the Visual Navigation Transformer (ViNT). ViNT offers a single, generalist framework that can solve a variety of navigation problems with little fine-tuning, in contrast to earlier specialized approaches. ViNT's adaptability makes it a fundamental tool for creating autonomous systems that are both broadly capable and flexible.

Reference

D. Shah, "VisualNav-Transformer: General-purpose goal-conditioned visual navigation models," GitHub repository, 2023. [Online]. Available: https://github.com/robodhruv/visualnav-transformer. [Accessed: Dec. 19, 2024].