Vision-based Navigation Models for Autonomous Rovers: Experimental Analysis and Comparison

Szilard Molnar and Levente Tamas

Automation Department, Technical University of Cluj-Napoca, Romania

***Abstract*— Traditional navigation algorithms rely heavily on the fusion of heterogeneous data from different sensor modalities for precise localisation and navigation tasks. However, these approaches lack generalisation and interaction with human operators, particularly in navigating unknown environments and human language-based task specifications. In contrast, the foundation models trained on internet-scale data proved higher generalisation capabilities and show an emerging trend in zero-shot learning for navigation tasks. Furthermore, these generic models can close the perception-planning loop through common sense reasoning, applicable to both language-based tasks and open-vocabulary visual recognition. However, due to the scarcity of robotics data for these vision-language-action-based approaches and the lack of clear evaluation protocols for these models, including safety guarantees, it is hard to evaluate the real benefits of these models. In this work, we analyse three different image-based navigation models: *ViNT*, *NoMaD*, and *GNM*. We propose specific metrics to quantitatively evaluate the visual-action-based navigation methods in indoor, outdoor, and simulation environments. In each scenario, these methods are adapted to a different robotic setup than the one provided in the original papers, providing an opportunity to benchmark the generalizability of these methods. The test setups and the associated codebase are available on the project webpage [1].**

I. Introduction

Standard *machine learning* and *computer vision* applications for autonomous rovers are usually focused on a single environment with a well-defined task in a specific use case. Additionally, the perception and planning components are decoupled, each utilising specific methods and with limited human-robot interaction capabilities. With the appearance of transformer-based generalised models, these limitations were relaxed. *Large language models* (LLMs) [1], [2] and *Visual Transformers* (ViTs) [3], are the most important components of these generalised models, which are trained on internet-scale images. The *vision language models* (VLMs) interpret both visual and linguistic aspects of the environment. Such VLMs are suitable for scene understanding, which requires the adoption of other modules for robotic applications. *Vision language action* (VLA) [4] models are more generalised for applications, which perceive the environment, receive a command prompt for the task, and then generate an action that can be performed by an agent (e.g., a robotic arm or other vehicle), closing the perception-planning loop. While VLAs refer to more generalised action models, in most cases, with the focus on robotic arms, moving robots can fall into the *vision language navigation* (VLN) [5] category.

In VLN, the task can be defined by an entire paragraph, as the navigation task is performed in multiple steps over time; hence, most VLN models also have a memory or history module. Both VLA and VLN return the action as a general command, which must be interpreted for each autonomous agent individually. Controlling a robotic arm using VLAs typically means that the VLA itself returns the next goal pose for the end effector; then, the inverse kinematics must be calculated for this pose. For navigation, the common action is a direction (e.g. *forwards*, *left*, *right*, *stop*), followed by a numerical value for distance or time. Therefore, the models themselves are agnostic to the possible joint angles, Cartesian positions, or motor torques, as they are specific to each use case.

In this work, we (1) provide an overview of vision-based navigation models that are available for testing with existing (or upcoming) open-sourced code and pretrained models, including VLA and VLN variants. (2) We quantitatively compare 3 vision-based methods for waypoint following applications in indoor, outdoor, and simulated environments, each with its own specific robot setup. The analyzed methods are: *Visual Navigation Transformers* (***ViNT***) [6], *Navigation with Goal Masked Diffusion* (***NoMaD***) [7], and *General Navigation Model* (***GNM***) [8]. Finally, (3) we introduce specific qualitative and quantitative measures for computing the distance between trajectories and measuring the success rate for each method. In the indoor environment, the endpoint accuracy is given by the *OptiTrack* localisation system, which gives both the distance and the angle errors of the robot. This comparison is performed for the chosen methods in three a priori unknown environments and hardware agnostic setups, which indicates the generalizability of the analysed methods.

II. Vision-based action models

General VLA methods can be used for various tasks, from robotic arm manipulation to navigation. Yang et al. [4] study the effect of including heterogeneous embodiments (i.e., increasingly different domains, such as manipulation and navigation) in a single model. Similarly, *Crossformer* [9]

This paper was financially supported by the Romanian National Authority for Scientific Research, project nr. PN-IV-P7-7.1-PTE-2024-0105 and by the Technical University of Cluj-Napoca for PhD students. Corresponding author: Levente.Tamas@aut.utcluj.ro

[1] https://github.com/molnarszilard/visualnav-transformer

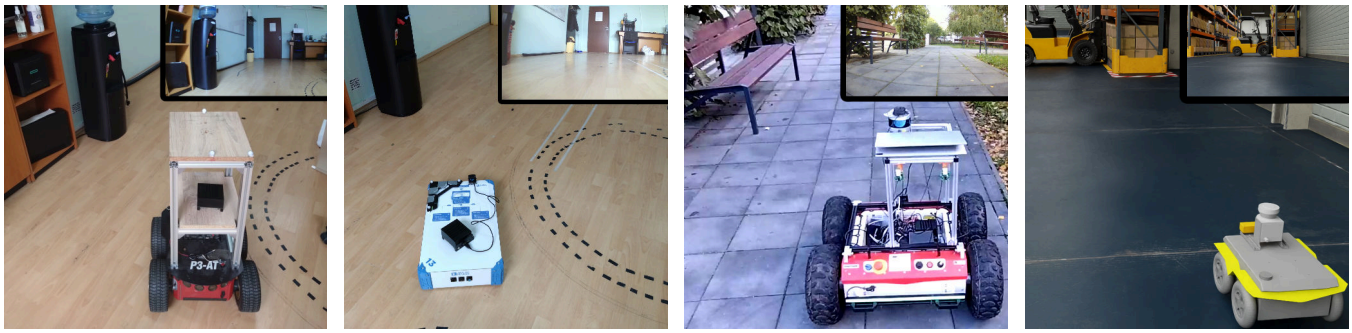

Fig. 1. The robot setups in the three environments (2 indoor, 1 outdoor, and 1 simulation), and what the camera sees (top-right corner).

offers multiple possible tasks, such as robotic arm manipulation, bi-manual manipulation, ground robot navigation, and unmanned aerial vehicle navigation, all using the same model. Each action type is associated with a set of tokens, and the corresponding output embeddings are then passed to the specific heads. π_0 [10] and $\pi_{0.5}$ [11] are more general VLA methods, which can simultaneously navigate a robot in an indoor environment, while also perform bi-manual tasks.

For VLNs, one of the most foundational works is the *Vision-and-Language Navigation in Continuous Environments* (*VLN-CE*) [12], which is a task set of continuous environments for vision language navigation methods. Domain-specific methods are encouraged to compete in the *VLN-CE* Challenge based on these datasets and task sets. From this challenge, multiple VLN methods emerged, two of which are *NavGPT* [13] and *NavGPT-2* [14]. These methods interact with the environment, language guidance, and navigation history to estimate a navigation action. The LLM part is based on *GPT-4* [15] and *BLIP-2* [1]. Another significant VLN model is the *NaVid* [16] method, which is entirely based on RGB images, without any prior maps, odometry, or depth data.

Anderson et al. [17] experiment with *Sim2Real* capabilities of implementing a VLN trained, then using a *Robot Operating System* (ROS)-based navigation stack [18] and *Simultaneous Localization and Mapping* (SLAM) [19]. Xu et al. [20] experiments with similar *Sim2Real* goals; however, they focus more on a narrow-angle RGB image and an online mapper. [21] is another method focusing on *Sim2Real*, but it only uses a narrow-angle monocular camera on the real robot, whereas the model was trained using a panoramic view of the simulation.

Visual Navigation Transformer (***ViNT***) [6] is a foundation model for navigation, based on attention transformer architecture [22]. This method presents an RGB image-based waypoint follower. The current observation and the goal observation are processed separately from the previous 5 observations on separate branches. The visual observations are encoded using *EfficientNet-B0* [23] to generate input tokens. These tokens are decoded with *Self-Attention Transformers* [22]. The resulting normalized actions are used to send navigation commands to any robot. In unseen environments, an image-to-image diffusion policy is used [24]. Similar methods are *Navigation with Goal Masked Diffusion* (***NoMaD***) [7] and ***GNM*** [8]. While the architecture of the ***NoMaD*** is similar to the architecture of ***ViNT***, ***GNM***, on the other hand, is a simple MobileNetv2-based encoder network [25], trained on a multirobot dataset to achieve better generalizability over multiple robots.

III. Method

A. Setup

Until now, we have presented a theoretical overview of different visual navigation methods for various types of robots. In the following part, we aim to present practical examples of these methods. This paper presents the first category of experiments, which is waypoint navigation based on real-time camera images. Therefore, the methods chosen for showcase and evaluation are: ***ViNT*** [6], ***NoMaD*** [7], and ***GNM*** [8]. As stated previously, these methods offer the possibility of performing image-based waypoint navigation and wandering. The original work presents a real-world setup with a *LoCoBot* robot and a *Jetson Orin Nano* computer. The robot is equipped with a *Kobuki* base and an *Intel D435* camera. All of this operation is based on the *Robotic Operating System* (ROS) [18], which connects all the different parts in a single, easily manageable environment.

In order to evaluate the hardware-agnostic capabilities of the methods, we experimented with 4 different setups, with different robot bases. We apply 2 indoor bases (*Pioneer3-AT* and *AD-R1M*), one outdoor base (*Husarion Panther*), and one simulated base in Isaac Sim (*Clearpath Robotics Jackal*). In all scenarios, the image processing is done using a *Jetson Xavier AGX* computer. Apart from adapting the input-output interfaces to our equipment, the source code is unmodified for the real-world scenarios, especially the image processing nodes, which provide a limited set of tunable parameters to optimize, meaning that our tests were as out-of-the-box as possible. In the simulator, for an optimal experiment, we needed to slightly modify the control algorithm, which converts the waypoint coordinates into commands. Additionally, the Husarion Panther and Isaac Sim require ROS2 commands; hence, a ROS1-Bridge was used there. The AGX module was set up with a power limit of 30 W (maximum setting) supplied using an external power bank.

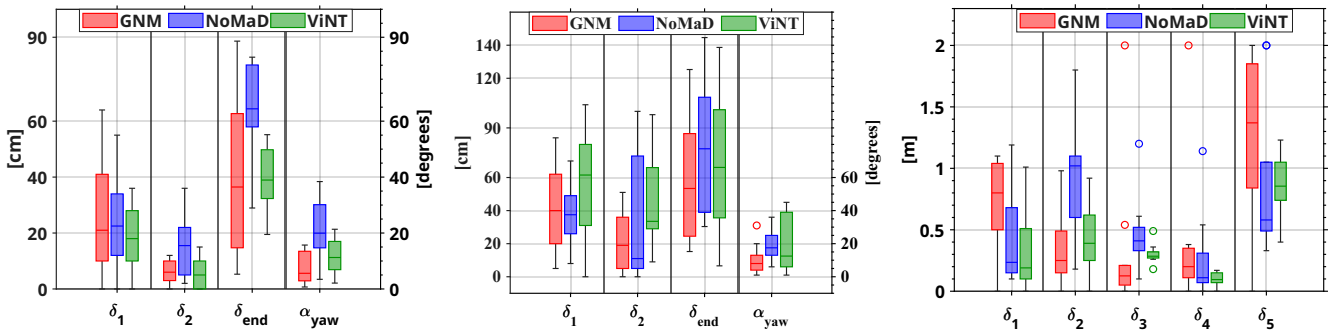

Fig. 2. The displacement errors of the 3 methods (***GNM*** [8], ***NoMaD*** [7], ***ViNT*** [6]). Left: indoor with Pioneer 3-AT, Center: indoor with AD-R1M, Right: outdoor with Husarion Panther.

B. Metrics

In [6], the authors present two different real-world applications. Since the accuracy of an exploration task is less measurable, we are experimenting with the image-based waypoint following task using our setup in indoor (laboratory), outdoor (park), and simulated (Isaac Sim warehouse) environments to verify the capabilities, particularly the generalizability, of the out-of-the-box models presented in [6]–[8].

In each test, the first step was to manually drive the robots along a path while images were captured. Then, these images were converted into topological maps for the methods. Then we performed the waypoint following 10 times for each of the 3 models. To ensure comparability between changing conditions, the methods were run in an alternating order.

The length of the indoor path was approximately $7\,\mathrm{m}$ (see Figure 3). Along this path, we marked three significant points: two in the middle and one at the end. In each run, we measured the closest distance of the robot from the two middle points using a measuring tape ($\delta_{1,2}$). The endpoint can give us a more precise measurement because, in our laboratory, we have an *OptiTrack* localisation system, which can provide us with the location and orientation of the robot to a millimetre-scale precision (δ_{end}).

In the outdoor tests, the path length was approximately $30\,\mathrm{m}$, along which a total of 5 significant points were marked. Due to the lack of the *OptiTrack* system and precise GPS, we only used a tape to measure the distance of the robot from the waypoints ($\delta_{1,2,3,4,5}$), and the robot's odometry for the distance travelled (Δ) and the duration.

In the Isaac Sim simulator, we used a predefined warehouse asset, where the length of the reference trajectory was $73\,\mathrm{m}$. Given the precise odometry in each run, we could calculate the exact trajectories. In this case, the metric can be defined by the Euclidean distance between the trajectories [26], without defining significant points.

We also measured the Success Rate (SR) and Conditional Success Rate (CSR) [27]. In our case, SR represents how often the robots managed to get to the end without manual intervention (Eq. 1), while CSR refers to the distance to each waypoint being less than a threshold ($0.5\,\mathrm{m}$ for indoor, and $1\,\mathrm{m}$ for outdoor (Eq. 2), these values refer to the approximate width of the respective robots, while CSR not applicable for the simulator due to the lack of waypoints). Manual intervention means that if the robot operator observes that the robot is about to hit an object, they grab (or control) the robot and rotate it in the direction of the next waypoint.

$$SR = \frac{1}{N}\sum_{i=0}^{N} s_i \tag{1}$$

$$CSR = \frac{1}{N \times M}\sum_{j=0}^{M}\sum_{i=0}^{N} (\delta_{i,j} < Th) \tag{2}$$

where N is the number of experiments for the given method, s describes if the given experiment was successful, M is the number of waypoints in each experiment, while δ is the measured distances from the reference waypoints. The Th is the threshold value ($0.5\,\mathrm{m}$ and $1\,\mathrm{m}$, for indoor and outdoor respectively).

IV. Results

A. Indoor

Starting with the results of the indoor tests: the measured displacement and orientation errors of the runs can be found in Figure 2. In these figures, we present the distance of the robot from the 3 waypoints, and also the yaw angle difference at the last waypoint (the pitch and roll value differences are not significant, and are not relevant since they are mostly caused by the uneven surface).

Numerical data can be found in Table I for the Pioneer 3-AT platform, while Table II shows numerical data for the AD-R1M platform. In Figure 3, the distribution of finish points is also presented. For Pioneer 3-AT, looking at the average values, for the two intermediary points, the ***ViNT*** model outperforms the other two, while at the last waypoint, it is slightly behind ***GNM*** in both distance and rotation. Other than a few outliers, ***NoMaD*** performs the worst in image-based waypoint navigation tasks, both in robustness and accuracy. These values show that while the Pioneer 3-AT robot performed the best with ***ViNT*** and ***GNM***, the AD-R1M platform prefers ***GNM*** and ***NoMaD***.

With the Pioneer 3-AT robot base, the SR shows the number of runs without interventions: ***GNM***-80 %, ***NoMaD***-30 %, ***ViNT***-100 %. CSR shows the number of waypoints,

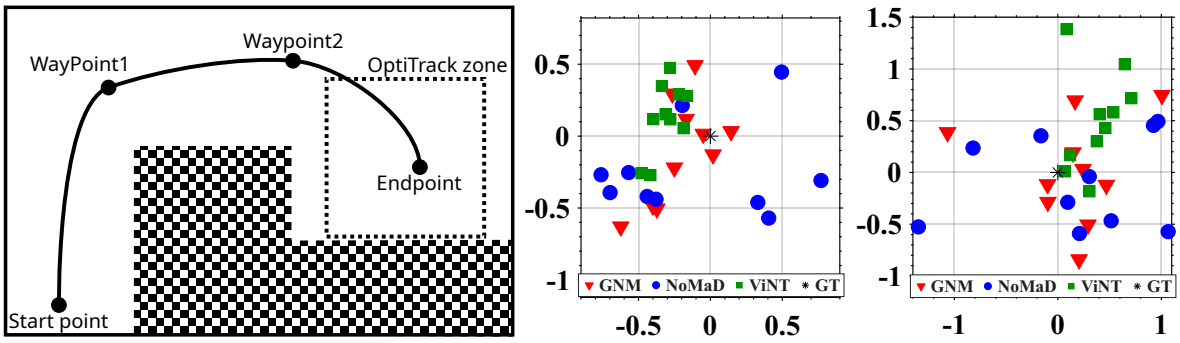

Fig. 3. Left: schematic of the ground truth path for the image-based waypoint following task. The total path is about 7 m long, and ends in an area where we can measure the robot using an *OptiTrack* localization system. (The checkerboard pattern represents an occupied zone as well as the walls of the drawing.) Right: the distribution of results of the endpoints for the analysed methods (measurements in m), first, for the Pioneer 3-AT, then for the AD-R1m robot.

TABLE I

NUMERICAL COMPARISONS OF THE INDOOR (WITH PIONEER 3 -AT) EXPERIMENTS FOR THE 3 MODELS: ***GNM*** [8], ***NoMaD*** [7], ***ViNT*** [6]. δ VALUES ARE IN *cm*, WHILE THE α_{yaw} VALUES ARE IN DEGREES.

Minimum	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	**0.0**	**0.0**	**5.3**	**0.7**
NoMaD	**0.0**	2.0	29.0	3.5
ViNT	**0.0**	**0.0**	19.5	2.1
Average	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	26.9	6.5	**39.1**	**7.6**
NoMaD	23.2	16.6	64.7	20.9
ViNT	**17.7**	**6.0**	40.3	11.2
Median	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	21.0	6.0	**36.5**	**5.6**
NoMaD	22.5	15.6	64.4	20.0
ViNT	**18.0**	**5.0**	38.9	11.3
StDev	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	20.6	**4.1**	26.9	**5.7**
NoMaD	16.6	11.9	15.9	11.7
ViNT	**11.9**	5.6	**11.7**	6.5
Maximum	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	64.0	**12.0**	88.6	**15.7**
NoMaD	55.0	36.1	82.9	38.4
ViNT	**36.0**	15.0	**55.8**	21.4

TABLE II

NUMERICAL COMPARISONS OF THE INDOOR EXPERIMENTS FOR THE 3 MODELS WITH THE AD-R1M ROBOT BASE: ***GNM*** [8], ***NoMaD*** [7], ***ViNT*** [6]. δ VALUES ARE IN *cm*, WHILE THE α_{yaw} VALUES ARE IN °.

Minimum	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	5	**0**	15.3	**1**
NoMaD	8	0	30.4	6
ViNT	**0**	9	**6.6**	**1**
Average	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	43.2	**21.6**	**59.7**	**10.6**
NoMaD	**37.6**	32.4	79.4	19
ViNT	54.9	43	68.5	18.4
Median	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	40	19	**53.4**	**8**
NoMaD	**37.5**	**11**	77.4	17.5
ViNT	61.5	33.5	66.1	12.5
StDev	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	26.4	**20**	**38.8**	**9.2**
NoMaD	**18.3**	37	39.7	10.2
ViNT	34	28.7	43.1	16.4
Maximum	δ_1	δ_2	δ_{end}	α_{yaw}
GNM	**84**	**51**	**125.3**	**31**
NoMaD	70	100	144.6	36
ViNT	104	98	138.7	45

which were closer to the robot than 0.5 meter: ***GNM***-83 %, ***NoMaD***-67 %, ***ViNT***-93 %.

With the AD-R1M robot base, the SR shows the number of runs without interventions: ***GNM***-30 %, ***NoMaD***-%, ***ViNT***-40 %. CSR shows the number of waypoints, which were closer to the robot than 0.5 meter: ***GNM***-67 %, ***NoMaD***-60 %, ***ViNT***-50 %.

B. Outdoor

Outdoor, the SR shows the number of experiments without interventions: ***GNM***-90 %, ***NoMaD***-0 %, ***ViNT***-70 %. CSR shows the number of waypoints, which were closer to the robot than 1 meter: ***GNM***-78 %, ***NoMaD***-76 %, ***ViNT***-92 %. Table III shows a more in-depth comparison of the numbers. To sum up the table, ***ViNT*** and ***GNM*** were close in terms of distance travelled, although both of them tended to stop before reaching the endpoint, hence the shorter Δ than in the case of the reference. While ***GNM*** once stopped before reaching the third waypoint (which is a significant outlier), it tended to give more powerful commands, hence its average duration time is the smallest (although this also indicates higher power consumption). ***NoMaD*** performed the worse in all metrics. In Figure 2, we capped the maximum values to 2 m, which refers to cases of extreme distances (***GNM*** once stopped even before the third waypoint, while ***NoMaD*** did not always stop at the endpoint).

C. Simulation

In this section, we present the results of the simulation runs. The SR shows the number of experiments without interventions: ***GNM***-0 %, ***NoMaD***-0 %, ***ViNT***-100 %. The average Euclidean trajectory distances are: ***GNM***-2.44 m, ***NoMaD***-2.03 m, ***ViNT***-0.82 m. The trajectories, compared to the reference, can be seen in Figure 4, while an in-depth numerical analysis can be seen in Table IV. From the experiments, we conclude that while ***GNM*** performs comparably to ***ViNT*** in real-world scenarios, ***GNM*** drops behind ***NoMaD*** in the simulated environment, where ***ViNT***

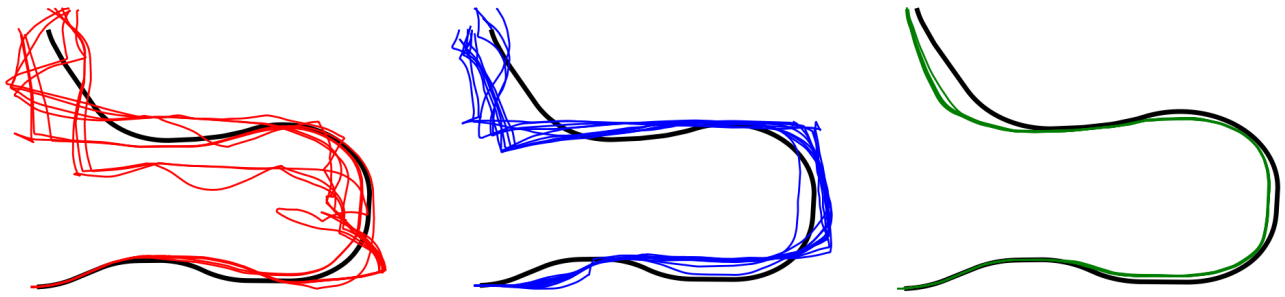

Fig. 4. The trajectories compared to the reference trajectory in simulation for several runs (black). Left: ***GNM*** (red), Center: ***NoMaD*** (blue), Right: ***ViNT*** (green).

TABLE III

NUMERICAL COMPARISONS OF THE OUTDOOR EXPERIMENTS FOR THE 3 MODELS: ***GNM*** [8], ***NoMaD*** [7], ***ViNT*** [6]. δ AND Δ VALUES ARE IN m, WHILE THE TIME IS MEASURED IN SECONDS. THE REFERENCE DISTANCE IS 29.5 m AND 106 s. THE VALUES MARKED WITH * ARE MEASURED WHEN THE METHOD DID NOT COMPLETE THE PATH, WHILE THE OPTIMAL δ IS THE ONE CLOSEST TO THE REFERENCE DISTANCE; OTHERWISE, THE MINIMAL VALUES ARE PREFERRED.

Minimum	δ_1	δ_2	δ_3	δ_4	δ_5	Δ	Time
GNM	**0**	**0**	**0**	**0**	**0**	13.1*	**190***
NoMaD	0.1	0.18	0.10	0	0.33	**29.7**	735
ViNT	**0**	**0**	0.18	**0**	0.4	28.2	569
Average	δ_1	δ_2	δ_3	δ_4	δ_5	Δ	Time
GNM	0.72	**0.31**	0.63	1.17	2.51	27.21	**387**
NoMaD	0.44	0.94	0.48	0.26	1.47	34.75	828
ViNT	**0.32**	0.42	**0.3**	**0.09**	**0.85**	**28.91**	585
Median	δ_1	δ_2	δ_3	δ_4	δ_5	Δ	Time
GNM	0.8	**0.25**	**0.13**	0.2	1.37	28.3	**401**
NoMaD	0.24	1.02	0.41	0.11	**0.58**	33.35	808
ViNT	**0.19**	0.39	0.29	**0.1**	0.86	**28.8**	586
StDev	δ_1	δ_2	δ_3	δ_4	δ_5	Δ	Time
GNM	0.36	0.29	1.54	3.11	4.44	5.1	70
NoMaD	0.37	0.48	0.29	0.35	1.87	4.6	76
ViNT	**0.33**	**0.26**	**0.08**	**0.06**	**0.26**	**0.6**	**13.1**
Maximum	δ_1	δ_2	δ_3	δ_4	δ_5	Δ	Time
GNM	1.1	0.98	5*	10*	15*	30.3	**429**
NoMaD	1.19	1.8	1.2	1.14	5*	43.8*	960*
ViNT	**1.01**	**0.92**	**0.49**	**0.17**	**1.23**	**30**	608

TABLE IV

NUMERICAL COMPARISONS OF THE SIMULATED EXPERIMENTS FOR THE 3 MODELS: ***GNM*** [8], ***NoMaD*** [7], ***ViNT*** [6]. THE REFERENCE LENGTH OF THE TRAJECTORY IS 73.34 m, WHILE THE DURATION IS 99 s. THE VALUES MARKED WITH * ARE MEASURED WHEN THE METHOD DID NOT COMPLETE THE PATH.

Minimum	Δ [m]	Time [sec]
GNM	78.04	125
NoMaD	78.95	143
ViNT	**74.06**	**122**
Average	Δ [m]	Time [sec]
GNM	86.57	163
NoMaD	83.65	153
ViNT	**74.22**	**125**
Median	Δ [m]	Time [sec]
GNM	87.5	165
NoMaD	82.62	153
ViNT	**74.22**	**125**
StDev	Δ [m]	Time [sec]
GNM	5.98	25
NoMaD	2.87	6
ViNT	**0.1**	**2**
Maximum	Δ [m]	Time [sec]
GNM	98.91	201*
NoMaD	87.1*	162
ViNT	**74.42**	**127**

TABLE V

THE PERFORMANCE OF THE MODELS, INCLUDING THE AVERAGE USED MEMORY, THE RUNTIME, AND THE POWER CONSUMPTION.

	Memory [GB]	Inference [sec]	Power [W]
GNM	**0.63**	**0.28**	**11.4**
NoMaD	1.29	1.16	12.7
ViNT	0.95	0.65	12.1

is precise and accurate. This result indicates that while ***ViNT*** works well in various environments, ***NoMaD*** is generally underperforming at the same level, regardless of the environment, and ***GNM*** shines in domain transfer between two real scenarios, but struggles when the domain shift is between real and simulated environments.

D. Performance

The performance of the models is independent of the robotic platform, since all tests used the *Jetson Xavier AGX* with the same camera. We measure the memory utilization, power consumption, and the inference time for each method executing them on the *Jetson Xavier AGX*. The values can be seen in Table V. The *Jetson Xavier AGX* was set to the maximum power mode available (MAXN - 35W), from an external battery over USB-C PD, while an internal software was monitoring the power consumption (the baseline power consumption on the whole device is 5.4 W). From every metric, ***GNM*** is the most efficient method, while ***NoMaD*** is not only the worst in quality but also in efficiency. ***ViNT*** is between them.

V. CONCLUSION

Recent trends show the applicability and availability of Vision Language Action and Vision Language Navigation models. In this work, we presented a few of these methods while conducting a quantitative comparison of three

methods, focusing on image-based waypoint following. Our tests revealed a significant difference between models, indicating that pre-trained models are not as generalizable as their counterparts. In our tests, ***ViNT*** performs well, while ***NoMaD*** is significantly more prone to errors in each testing environment. On the other hand, ***GNM*** performs well in real-world environments, while the results from the simulated environments are the worst. Even on a relatively short path, the distance error can be more than $10\,\%$ of the length of the traversed trajectory. This refers to the variability of the capability to generalize the learned environment and apply it to new ones.

Overall, the applicability of such models is increasing, but there is still a place for improvement, as the tested models can cause collisions. In the future, these waypoint follower methods should be compared to VLN methods.

REFERENCES

[1] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023, pp. 19 730–19 742.

[2] H. Touvron, L. Martin, K. Stone, P. Albert *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *Computing Research Repository*, vol. abs/2307.09288, 2023.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[4] J. H. Yang, C. Glossop, A. Bhorkar, D. Shah *et al.*, "Pushing the Limits of Cross-Embodiment Learning for Manipulation and Navigation," in *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, D. Kulic, G. Venture, K. E. Bekris, and E. Coronado, Eds., 2024.

[5] P. Anderson, Q. Wu, D. Teney, J. Bruce *et al.*, "Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3674–3683.

[6] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz *et al.*, "ViNT: A Foundation Model for Visual Navigation," in *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 2023, pp. 711–733.

[7] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration," in *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*. IEEE, 2024, pp. 63–70.

[8] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A General Navigation Model to Drive Any Robot," in *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*. IEEE, 2023, pp. 7226–7233.

[9] R. Doshi, H. R. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 2024, pp. 496–512.

[10] K. Black, N. Brown, D. Driess, A. Esmail *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," *Computing Research Repository*, vol. abs/2410.24164, 2024.

[11] P. Intelligence, K. Black, N. Brown, J. Darpinian *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *Computing Research Repository*, vol. abs/2504.16054, 2025.

[12] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12373. Springer, 2020, pp. 104–120.

[13] G. Zhou, Y. Hong, and Q. Wu, "NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 7641–7649.

[14] G. Zhou, Y. Hong, Z. Wang, X. E. Wang, and Q. Wu, "NavGPT-2: Unleashing Navigational Reasoning Capability for Large Vision-Language Models," in *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VII*, ser. Lecture Notes in Computer Science, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., vol. 15065. Springer, 2024, pp. 260–278.

[15] OpenAI, "GPT-4 Technical Report," *Computing Research Repository*, vol. abs/2303.08774, 2023.

[16] J. Zhang, K. Wang, R. Xu, G. Zhou *et al.*, "NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation," in *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, D. Kulic, G. Venture, K. E. Bekris, and E. Coronado, Eds., 2024.

[17] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar *et al.*, "Sim-to-Real Transfer for Vision-and-Language Navigation," in *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. J. Tomlin, Eds., vol. 155. PMLR, 2020, pp. 671–681.

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust *et al.*, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[19] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[20] C. Xu, H. T. Nguyen, C. Amato, and L. L. S. Wong, "Vision and Language Navigation in the Real World Via Online Visual Language Mapping," *Computing Research Repository*, vol. abs/2310.10822, 2023.

[21] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Sim-to-Real Transfer via 3D Feature Fields for Vision-and-Language Navigation," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 2024, pp. 2982–2995.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[23] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 6105–6114.

[24] C. Saharia, W. Chan, H. Chang, C. Lee *et al.*, "Palette: Image-to-Image Diffusion Models," in *ACM SIGGRAPH 2022 Conference Proceedings*, ser. SIGGRAPH '22. Vancouver, BC, Canada: Assoc. for Computing Machinery, 2022, pp. 1–10.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[26] J. M. Phillips and P. Tang, "Simple Distances for Trajectories via Landmarks," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019, Chicago, IL, USA, November 5-8, 2019*, F. B. Kashani, G. Trajcevski, R. H. Güting, L. Kulik, and S. D. Newsam, Eds. ACM, 2019, pp. 468–471.

[27] X. Song, W. Chen, Y. Liu, W. Chen *et al.*, "Towards Long-Horizon Vision-Language Navigation: Platform, Benchmark and Method," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*. Computer Vision Foundation / IEEE, 2025, pp. 12 078–12 088.