2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Synthetic vs. Real Training Data for Visual Navigation

Lauri Suomela[1], Sasanka Kuruppu Arachchige[1], German F. Torres[1],
Harry Edelman[2], Joni-Kristian Kämäräinen[1]
[1]Tampere University [2]Turku University of Applied Sciences

***Abstract*— This paper investigates how the performance of visual navigation policies trained in simulation compares to policies trained with real-world data. Performance degradation of simulator-trained policies is often significant when they are evaluated in the real world. However, despite this well-known sim-to-real gap, we demonstrate that simulator-trained policies can match the performance of their real-world-trained counterparts. Central to our approach is a navigation policy architecture that bridges the sim-to-real appearance gap by leveraging pretrained visual representations and runs real-time on robot hardware. Evaluations on a wheeled mobile robot show that the proposed policy, when trained in simulation, outperforms its real-world-trained version by 31 and the prior state-of-the-art methods by 50 points in navigation success rate. Policy generalization is verified by deploying the same model onboard a drone. Our results highlight the importance of diverse image encoder pretraining for sim-to-real generalization, and identify on-policy learning as a key advantage of simulated training over training with real data. Code, model checkpoints and multimedia materials are available at lasuomela.github.io/faint.**

I. Introduction

Recently, learning-based visual navigation methods have received attention as potential replacements for traditional sense-plan-act approaches that leverage geometric environment representations. Navigation systems with learned components have many advantages, for example, being able to utilize semantic information to infer traversability [5, 24] and guide exploration [14, 34], and incorporate other task specifications in addition to metric coordinates [1]. The primary robot learning paradigms involve imitation learning (IL) from real-world robot datasets [24, 45, 47, 48, 52, 60], and reinforcement learning (RL) or imitation from scripted [12, 29] or human experts [14, 38] in simulation [13, 19, 28, 57]. The performance of learned robot policies heavily depends on the quality, quantity, and diversity of training data [21]. Real and synthetic data collection each have distinct strengths and weaknesses with respect to these factors. Real-world data collection is labor-intensive and platform-specific, yet it reduces domain shifts at policy deployment, while synthetic data generation offers greater scalability and flexibility but faces the *simulation-to-real* (sim2real) gap [22], limiting generalization to real environments.

Our work investigates how navigation performance differs between policies trained with real-world data and those trained entirely in simulation. Although prior works [14, 23, 41, 50] have assessed the sim2real gap by evaluating simulation-trained policies in simulated and real environments, they have not directly compared these policies with those learned from real-world data. We address this gap by evaluating whether simulation-only training can achieve performance competitive with real-world data.

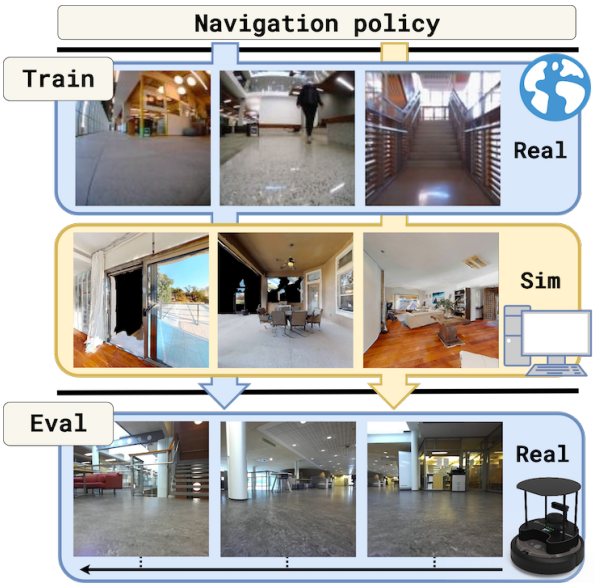

Fig. 1: We investigate how simulation-trained navigation policies compare to ones trained with real-world data when deployed on a real robot.

Previously proposed navigation policies have been tailored for either synthetic or real-world data. To facilitate our experiments, we introduce a novel policy architecture for visual topological navigation: **F**ast **A**ppearance-**I**nvariant **N**avigation **T**ransformer (FAINT). FAINT can be trained on either real or simulated data, is lightweight enough for deployment on resource-constrained robot hardware, and demonstrates robust sim-to-real transfer. We deployed sim and real-data-trained versions of the policy on a wheeled mobile robot and a drone, and performed more than 50 hours of evaluation in challenging real-world indoor environments.

Our findings demonstrate that a navigation policy trained entirely in simulation can perform on par with—or even outperform—those trained on real-world data. Simulation-trained FAINT surpasses its real-world-trained counterpart by 31 points and the previous state-of-the-art by 50 points in navigation success rate. Despite being trained with fully synthetic data, the policy can successfully adapt to unseen environment conditions and different robot platforms. Based on these results, we identify scalable data generation and the ability to perform on-policy learning as key advantages of simulation over real-world data.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

II. Related work

Learning from real-world data. Most works formulate learning navigation from real-world demonstrations as a goal-conditioned imitation learning [6] problem. The main challenges are related to generalization across unseen environments and embodiments. Kahn et al. [24] train a navigation model in one of the first works to demonstrate generalization to novel environments. Shah et al. [47, 48, 52] achieve generalization across robot platforms by training topological navigation models with data collected from multiple different robots. Suomela et al. [54] improve upon this line of work by reducing the dependency on robot-originated training data through the use of place recognition models [32]. We build on these prior architectures, but introduce improvements that we find critical for sim2real generalization.

Learning from synthetic data. Simulation offers scalability and tailorability, making it attractive for training navigation policies. However, the sim2real gap [22] remains a major challenge for real-world deployment [14]. While policies trained on raw RGB inputs with domain randomization have shown some success, their generalization remains limited [33, 40, 64]. Training policies on sensor abstractions such as depth images [29, 58], segmentation masks [13, 34], and feature points [26] is a promising alternative. Recently, feature maps from *pre-trained visual representation* (PVR) models have proven effective for bridging the sim2real gap. Ehsani et al. [11, 12, 20, 62] use a frozen SigLIP [63] encoder for navigation and manipulation tasks, while Silwal et al. [50] demonstrate successful sim2real transfer in image-goal navigation using a VC-1 [31] encoder. However, these large PVR's are unsuitable for real-time deployment on resource-constrained robots. In this work, we show that even smaller, distilled PVR models [49] enable sim2real transfer while being suitable for on-robot execution.

Sim2Real investigations. Substantial amounts of work have been put into studying and quantifying the sim2real gap by comparing policies' performances in simulation and the real world [14, 23, 42, 57]. What we identify as still missing is a real-world comparison of the performance of navigation models trained with synthetic and real datasets. The work most similar to ours is the investigation by Silwal et al. [50], which examines the performance of manipulation policies trained with real and simulated data in real-world settings. They find that simulated policies trained with few-shot imitation learning exhibit poor sim2real transfer and underperform compared to real-data policies. However, they also train an image-goal navigation policy using large-scale reinforcement learning in simulation, which performs well in real-world environments. Notably, they do not compare the image-goal policy to a model trained on real-world data, likely because there are no suitable real-world datasets. In contrast, our investigation focuses on visual topological navigation, a task for which real-world datasets are available. This allows us to directly compare the policies trained on real and synthetic data for the same task.

III. Methods

In this Section, we present the definition of the navigation task (Sec. III-A), outline the architecture of the model used in the experiments (Sec. III-B), and describe the datasets and learning approach for synthetic (Sec. III-C) and real-world (Sec. III-D) data.

A. Problem formulation

Topological visual navigation. We perform the investigation in the context of topological image-goal navigation [43, 47, 54] because it is a navigation task with some of the most extensive real-world datasets available. Topological approaches divide a navigation route into a set of intermediate subgoals $\{s_0, s_1, \ldots, s_n\}$, each with an associated image observation. These subgoals comprise a topological map $\mathcal{M}$, created from images collected prior to robot deployment. During navigation, at each time step t a *subgoal selection policy* π_s finds the next subgoal s_t along the route to the final goal, and returns the corresponding subgoal image S_t:

$$S_t = \pi_s(O_t, \mathcal{M}) \tag{1}$$

where O_t is the current observation image. Given S_t and a sequence of P recent observations $\mathbf{O}_t = \{O_{t-P+1}, \ldots, O_t\}$, a *goal-reaching policy* π_g then produces a sequence of H robot control commands $\mathbf{a}_t = \{a_t, \ldots, a_{t+H-1}\}$ towards the subgoal:

$$\mathbf{a}_t = \pi_g(\mathbf{O}_t, S_t). \tag{2}$$

We adopt the subgoal selection method from Suomela et al. [54], and perform the selection with place recognition [32] models that can be trained with large-scale datasets from *e.g.* Google Streetview [2]. This lets us focus on the goal-reaching policies which are more tightly tied to the robot embodiment and for which the training data is scarce.

Action space. The goal-reaching policy outputs trajectory *waypoints* relative to the robot coordinate frame, as it allows embodiment-agnostic control and direct comparison to relevant prior methods [47, 48, 54]. Each waypoint $a \in \mathbf{a}_t$ is a pose $a = [x, y, \theta] \in SE(2)$ where (x, y) is the position and θ the orientation. During deployment, a simple PD-controller estimates velocity commands from the waypoints.

B. FAINT model architecture

We improve prior goal-reaching policies [48] by integrating a *pretrained visual representation* (PVR's) and a novel *binocular goal encoder*. FAINT has just 12M parameters—half the size of prior models [52]—enabling real-time inference on resource-constrained hardware. An overview of the architecture is shown in Figure 2. We describe each component in detail below.

Pretrained visual representation. As a key to bridging the sim-to-real appearance gap, we leverage image encoders pretrained on a diverse visual tasks. We adopt the $5M$-parameter *Tiny CDDSV* variant of the Theia encoder [49], which distills representations from CLIP [36], DiNOv2 [35], Depth Anything [61], Segment Anything [27], and ViT [10]. Despite its small size, it demonstrates robust sim-to-real

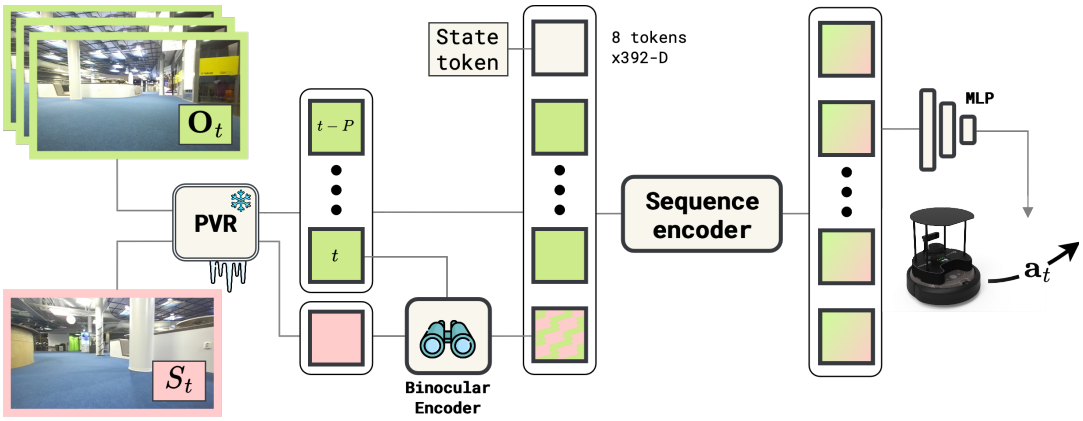

Fig. 2: **Model architecture.** FAINT implements the goal-reaching policy $\mathbf{a}_t = \pi_g(\mathbf{O}_t, S_t)$. Observation and subgoal images are encoded with a frozen PVR, and a binocular encoder refines the goal tokens by conditioning on the latest observation. A sequence encoder with a predictor head then produces the actions $\mathbf{a}_t$. Subgoals S_t are obtained from a separate subgoal selection policy π_s.

generalization. The weights are frozen during training to avoid overfitting.

Binocular encoder. Previous work [48, 53] has shown that conditioning the goal image on the current observation improves navigation performance, for instance by facilitating estimation of the relative pose between the robot and the goal [4]. However, standard methods that concatenate the observation and goal images along the channel dimension are incompatible with pretrained image encoders. Inspired by the binocular vision architecture of Weinzaepfel et al. [59], we utilize a transformer decoder to extract correspondences between the encoded observation and goal tokens. More specifically, the decoder alternates between self-attention on the goal tokens and cross-attention on the observation tokens. A key strength of this approach is its ability to learn navigation-relevant cues directly from arbitrary, frozen pretrained embeddings. As illustrated in Fig. 3, our binocular encoder identifies matches between image features despite being trained end-to-end with the rest of the policy, without explicit supervision. We use 4 transformer layers with 4 attention heads.

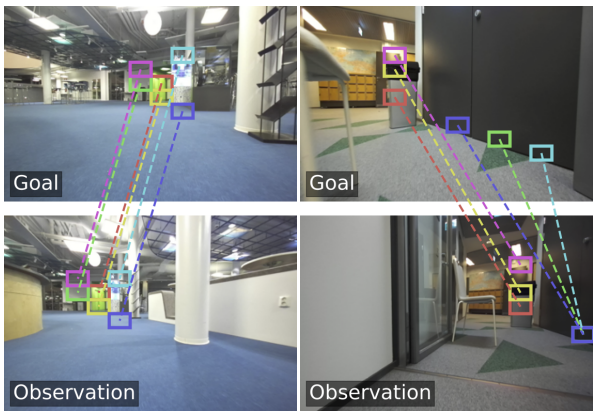

Fig. 3: Implicit correspondences of the six highest attention values in the binocular encoder's first cross-attention layer.

Sequence encoder. The observation and conditioned goal tokens are processed by a transformer encoder with non-causal self-attention. Before input into the sequence encoder, the patch tokens of each image are compressed into a one-dimensional vector with a 2D convolution layer followed by flattening, similar to [31]. A learnable state token is added to the sequence, and its corresponding token from the sequence encoder output is passed to a predictor head. This produces the final output, a sequence of waypoints $\mathbf{a}_t$. The sequence encoder consists of 4 layers with 4 attention heads each.

C. Learning to navigate in simulation

Learning goal-reaching from a shortest-path oracle. We train a learning-based agent to mimic a scripted oracle agent that follows the shortest paths between the episode start and goal locations. The oracle has privileged access to the simulator state and utilizes a proportional controller for path tracking. The student agent is trained as if the oracle was navigating a sequence of subgoals $s \in \mathcal{M}$ along the shortest path. The agent predicts the oracle actions $\mathbf{a}_{gt}$ while only having access to the $P = 6$ latest egocentric camera observations $\mathbf{O}_t$, and the subgoal image S_t captured at the next subgoal pose s_t. The predicted actions are trained to minimize the *mean squared error* loss $\mathcal{L} = MSE(\mathbf{a}_t, \mathbf{a}_{gt})$. The set of subgoals $\mathcal{M}$ is randomly sampled along the shortest path so that for each consecutive subgoal the geodesic distance $d(s_n, s_{n+1}) \in [d_{min}, d_{max}]$, where $d_{min} = 0.5\,\text{m}, d_{max} = 3.0\,\text{m}$ are the minimum and maximum subgoal separation. The subgoal at the time step t is chosen from $\mathcal{M}$ based on the distance from the position of the agent. At each time step, the oracle agent is rolled out for $H = 5$ steps to acquire future actions $\mathbf{a}_{gt}$ for each pair of observation and subgoal. Each triplet $(\mathbf{O}_t, S_t, \mathbf{a}_{gt})$, illustrated in Fig. 4, is saved to disk for use as training data for the student agent.

Data distribution. Naively imitating the oracle agent leads to poor performance, as shown in Sec. IV-D. During deployment, compounding prediction errors lead to covariate shift between training data and actual observations [51]. If a policy

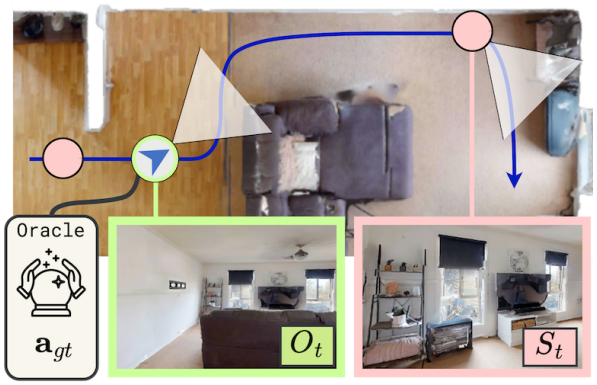

Fig. 4: Training data collected from the simulator - oracle actions $\mathbf{a}_{gt}$ that control the agent, agent observation O_t, and subgoal image S_t.

has only been trained with shortest-path trajectories, it might not be able to recover back after ending in a state outside the shortest-path distribution. Thus, we employ *DAgger* [39] to diversify the state distribution of the training data. During training data collection, the simulated agent executes the student policy action instead of $\mathbf{a}_{gt}$ with probability $p(\mathbf{a}_t) = \beta^r$, where β is a decay coefficient and r is the number of the current training round. The student action is only executed if it does not lead to a collision.

Simulator & training setup. Training was carried out in the Habitat [44] simulator with the train split of the HM3D environments and the PointNav route dataset [37]. We sampled routes where the agent can reach the goal within 500 steps without collisions. Agent radius was set to 0.1 m and its movement was simulated by kinematic control [57]. The RGB camera *field-of-view* (FOV) was set to 110° with resolution of 224×126. We trained models for 10 rounds of DAgger with $\beta = 0.8$, batch size 512, AdamW [30] optimizer and initial learning rate 2×10^{-4}, decayed with cosine schedule. Images were augmented with color jitter and posterization.

D. Learning from real-world data

The real-world-data version of FAINT was trained with the publicly available topological navigation datasets, specifically RECON [46], GoStanford [17], SACSoN [18], SCAND [25], and TartanDrive [56]. The synthetic part of GoStanford was omitted to avoid mixing the real and synthetic data. Trajectories, when sampled at 4 Hz, have $\sim 1.2M$ image frames. We follow the training procedure described by Shah et al. [48] with the difference that we omit the temporal distance prediction. To produce training data, pairs of observation and goal images are sampled from the dataset trajectories. After sampling a sequence of observations $\mathbf{O}_t$, a goal image S_t is picked randomly from the same trajectory, $[l_{min}, \ldots, l_{max}]$ frames in the future from t, similar to hindsight relabeling [15]. The H future poses $\mathbf{a}_{gt}$ relative to the current pose of the robot are then used as action labels for training. To enable learning across data from heterogeneous robots, the waypoints are normalized by the average waypoint distance of each dataset [47]. We used the same training setup as with simulated training, except a smaller batch size of 256 and initial learning rate 5×10^{-4}.

IV. Experiments

We conducted real-world navigation tests in various indoor environments in experiments designed to answer the following questions.

- **Q1**: How well can a policy perform when trained in simulation instead of with real-world data?
- **Q2**: How does FAINT trained with synthetic data compare to the previous state-of-the-art?
- **Q3**: How do different architectural choices affect the policy's sim-to-real generalization?
- **Q4**: Does training in simulation require the deployment embodiment to closely match the training embodiment?

A. General setup

Hardware. Experiments **Q1.-Q3.** were performed on a Turtlebot4 robot with a 110° FOV ZED 2i camera and an Nvidia Jetson Orin AGX, moving at 0.3 m/s. For studying **Q4.**, FAINT was deployed on a custom-built *Agipix*-drone, equipped with a forward-facing 110° FOV USB camera and a Jetson Orin NX. All computation was performed on board the robots.

Deployment. FAINT was deployed within the PlaceNav [54] framework, which divides navigation into separate goal-reaching and subgoal selection policies. Subgoal selection was performed with a ResNet18 [16] variant of the EigenPlaces [3] place recognition model followed by Bayesian filtering. The subgoal selection and goal-reaching policies run in separate threads, both at 4 Hz. To navigate a route, the robot is first teleoperated to capture map images. A new image is added to the topological map every 5 s, meaning node spacing of ∼1.5 m at robot speed of 0.3 m/s. During navigation, the robot follows the sequence of image goals from the beginning to the end of the route.

Evaluation. We evaluated navigation performance by average *success rate* (SR) [1]. Similar to [54], a real-world navigation episode is considered successful if the navigation system's subgoal selection module localizes to the last image of the topological map. An episode is unsuccessful if the robot collides with the environment or gets lost in such a way that it cannot return to the test route. In the simulator, we consider an episode successful if the agent arrives within 0.4 m of the goal within 500 simulator steps.

Test environments. The experiments were carried out in various indoor environments including a real apartment, offices, public spaces on a university campus, and a nuclear fallout shelter. The tests were limited to indoor environments to reduce the domain gap between the deployment and the simulated training environments. The test routes had different features relevant to navigation, illustrated in Figures 5 and 6. The lengths of the test routes ranged from 5 to 25 meters.

(a) Open space (b) Tight maneuver (c) Clutter

Fig. 5: Example segments from different types of test routes.

TABLE I: **FAINT** trained with **real vs. simulated** data, *success rates* (↑) over 3 repetitions of 7 routes per category.

Dataset	Samples	Open space $n = 21$	Tight maneuver 21	Clutter 21	Total 63
Real	$1.2M$	0.43	0.52	0.38	0.44
$\text{Sim}_{10\%}$	$1.2M$	0.57	0.05	0.05	0.22
Sim	$12.0M$	**0.86**	**0.62**	**0.76**	**0.75**

TABLE II: **SOTA comparison** *success rates* (↑) over 3 repetitions of 4 routes per category.

Method	Open space $n = 12$	Tight maneuver 12	Clutter 12	Illumination change 12	Total 48
NoMAD [52]	0.08	0.08	0.25	0.00	0.10
PlaceNav [54]	0.67	0.25	0.33	0.00	0.31
ViNT [48]	**0.92**	0.25	0.42	0.00	0.40
FAINT (ours)	**0.92**	**0.92**	**0.75**	**1.00**	**0.90**

B. Synthetic vs. real training data

To answer **Q1.**, we trained versions of FAINT with different amounts of real and synthetic data and compared the models' performances in the real-world. FAINT_{Real} was trained with behavior cloning (BC), the simulated ones with DAgger.

Results. Table I shows the experiment results divided by route category. When training with the same number of samples, real-world data produces better navigation performance than the synthetic data. On the 'open space'-routes, the performance is similar, but on the more challenging routes the $Sim10\%$ policy often collides as result of cutting too close to obstacles. A 10-fold increase in the amount of synthetically generated data, however, leads to a drastic performance increase. FAINT_{Sim} outperforms the other two by a wide margin in all categories. It is able to perform complex maneuvers to *e.g.* go around obstacles, and reach goal images not within the immediate camera view.

Some of the performance gap between $Real$ and Sim may be explained by dataset size—FAINT_{Real} trained on 12M samples might match FAINT_{Sim}. However, scaling simulated data is essentially free, while real-world data collection is very labor-intensive. We suggest on-policy learning as a more insightful explanation. FAINT_{Sim} trained with BC (see Sec. IV-D) performs poorly despite being trained with $12M$ samples, demonstrating that scale alone is not sufficient. Results in [7] show that DAgger can require a high number of expert queries to work properly. We hypothesize this to be a partial cause of the performance gap between $Sim10\%$ and Sim, not the difference in data scale per se. Interestingly, FAINT_{Real} exhibits similar failure modes as FAINT_{Sim} trained with BC, getting stuck in feedback loops such as spinning in place [51]. This parallel suggests that the behavior cloning's inability to handle compounding errors also affects real-world policies trained without on-policy corrections. This underscores the potential of simulation for robot learning, as large-scale on-policy learning is impractical in the real-world.

C. SOTA comparison

To study **Q2.**, we compared FAINT_{Sim} with topological navigation methods from previous work, trained with real-world data. We only considered methods that can run onboard the robot, which rules out larger models such as CrossFormer [9]. The baseline models were deployed with the author-provided model checkpoints.

We extend the experiment setup of Sec. IV-B with a new route type with illumination change between map collection and deployment. Additionally, we conducted a controlled study on illumination change in which we captured maps

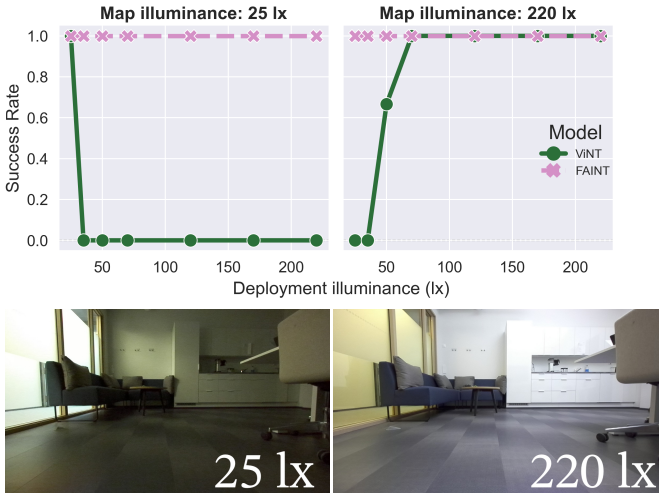

Fig. 6: ViNT and FAINT were tested under various illumination levels, with two maps captured under 25 lx and 220 lx.

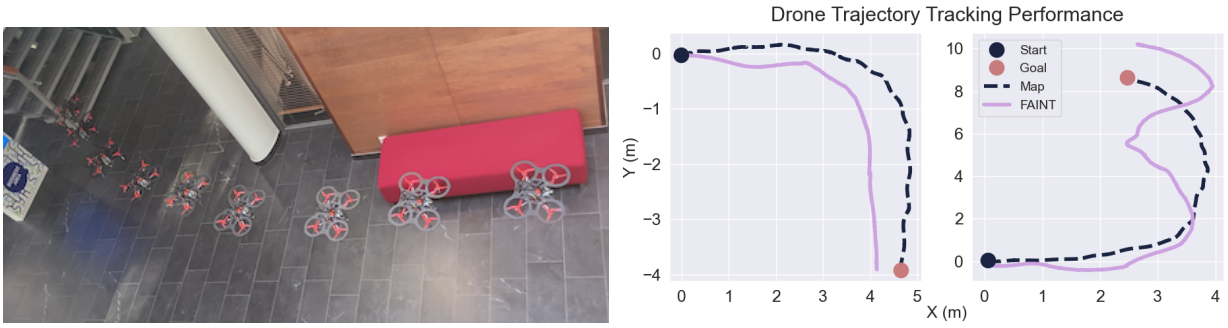

Fig. 7: Drone trajectory relative to $\mathcal{M}$ when controlled by FAINT. The drone successfully reached the goal on the left trajectory ($RMSE$ 1.07 m), but missed it on the right ($RMSE$ 1.42 m).

on one easy route under regular indoor lighting [8] (220 lx) and dim lighting (25 lx). During testing, illumination was progressively transitioned between these two levels.

Results of the SOTA comparison are shown in Table II. NoMAD performs surprisingly poorly. We hypothesize the diffusion-based method to be less robust to variation in cameras and embodiments. FAINT outperforms other methods across every route category. The baselines struggle with segments that require sharp turns, often failing to navigate effectively toward the goal. We hypothesize this is caused by the limited receptive fields of their convolutional goal-conditioning modules. In contrast, FAINT 's binocular encoder allows it to associate observation and goal image features even under wide baseline changes (see Fig. 3). The performance differences were the most drastic on routes with illumination change. The baselines struggle with even moderate changes, while FAINT's performance is consistent across illuminations. With controlled illumination change (Fig. 6), both ViNT and FAINT perform well when the deployment illumination is close to the reference, but as the difference grows, ViNT degrades and fails completely. With the 25 lx map, it only succeeds if the deployment illumination is the same. In contrast, FAINT succeeds under all conditions with both references. This robustness is likely due to FAINT 's use of a pre-trained visual encoder, which helps bridge appearance gaps — including those caused by lighting changes.

D. Simulation-to-real generalization

To address **Q3.** we analyze model design choices' effects on sim2real generalization. We trained models with $12M$ samples from simulation with either behavior cloning (BC) or DAgger. Each method was evaluated over 3 repetitions across 10 real-world routes, and on the 2500 routes of the HM3D Val split [37].

Results in Table III highlight the role of image encoder pretraining and on-policy data for sim2real transfer. The frozen EfficientNet [55] trained for classification on ImageNet transfers poorly, performing even worse than ViNT with an unfrozen encoder. The two Theia [49] variants distill representations from large models trained for diverse visual tasks, and enable strong real-world performance even without fine-tuning. Models trained with BC and DAgger perform similarly in simulation but differ greatly in real-world performance. We attribute this to the stronger prediction error compounding caused by the sim2real gap and higher non-determinism of the real world. DAgger exposes the policy to a wider state distribution during training, drastically improving real-world performance.

TABLE III: **Simulation-to-real** experiment *success rates* (↑). Note that ViNT does not allow freezing the encoder.

Method	Encoder type	Encoder frozen	Mode	Real $n = 30$	Sim 2500
FAINT	Theia CDDSV	✓	BC	0.23	0.87
	Theia CDDSV	✓	DAgger	**0.80**	**0.91**
	Theia CDIV	✓	DAgger	0.60	**0.91**
	EfficientNet-B0	✓	DAgger	0.13	0.79
ViNT [48]	EfficientNet-B0	✘	DAgger	0.40	0.87

E. Cross-embodiment generalization

To study **Q4.**, we trained FAINT with a simulated wheeled robot embodiment and deployed to a real drone without any modifications. The policy controlled the drone's forward velocity, up to 0.4 m/s, and yaw rate at a fixed elevation of ∼1.5 m. We tested drone navigation on two 'open space'-type routes. Fig. 7 shows the drone trajectories and tracking metrics compared to the reference routes. These preliminary results indicate that the deployment embodiment does not have to be strictly similar to the one used in training. We leave more thorough analysis of cross-embodiment generalization to future work.

V. Conclusion

This work demonstrated that a simulation-trained visual navigation policy can reach performance comparable to policies trained with real data. We proposed a novel navigation policy architecture that is similar to the previous state-of-the-art, but introduces key modifications that make it suitable for training with both real and synthetic data. Comparison of synthetic and real data trained versions of the policy show that on-policy learning is a major advantage of simulated training, providing robustness to covariate shift. The findings suggest combining off-policy real-world datasets with on-policy corrections from simulation as an interesting avenue for future work.

ACKNOWLEDGMENT

The authors wish to acknowledge CSC – IT Center for Science, Finland, for generous computational resources. The work was financially supported by the Technology Innovation Institute, and also received funding from the European Commission's HORIZON.1.2 - Marie Skłodowska-Curie Actions (MSCA) under Grant agreement No. 101072634, project RAICAM.

REFERENCES

[1] Anderson, P. et al. On Evaluation of Embodied Navigation Agents. *arXiv:1807.06757 [cs]*, July 2018. 1, 4

[2] Berton, G., Masone, C. and Caputo, B. Rethinking Visual Geo-localization for Large-Scale Applications. In *2022 IEEE/CVF CVPR*, June 2022. 2

[3] Berton, G. et al. EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition. In *2023 IEEE/CVF ICCV*, 2023. 4

[4] Bono, G. et al. End-to-End (Instance)-Image Goal Navigation through Correspondence as an Emergent Phenomenon. In *International Conference on Learning Representations (ICLR)*, October 2023. 3

[5] Castro, M.G. et al. How Does It Feel? Self-Supervised Costmap Learning for Off-Road Vehicle Traversability. In *2023 IEEE ICRA*, May 2023. 1

[6] Codevilla, F. et al. End-to-End Driving Via Conditional Imitation Learning. In *2018 IEEE ICRA*, May 2018. 2

[7] de Haan, P., Jayaraman, D. and Levine, S. Causal Confusion in Imitation Learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 5

[8] DiLaura, D.L. et al. *The lighting handbook: reference and application*. Illuminating Engineering Society of North America, New York, NY, 10th ed. edition, 2011. 6

[9] Doshi, R. et al. Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation. In *Conference on Robot Learning*. September 2024. 5

[10] Dosovitskiy, A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 2

[11] Eftekhar, A. et al. The One RING: a Robotic Indoor Navigation Generalist, December 2024. arXiv:2412.14401 [cs]. 2

[12] Ehsani, K. et al. SPOC: Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World. In *2024 IEEE/CVF CVPR*, 2024. 1, 2

[13] Geles, I. et al. Demonstrating Agile Flight from Pixels without State Estimation. In *Robotics: Science and Systems XX*, July 2024. 1, 2

[14] Gervet, T. et al. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, June 2023. 1, 2

[15] Ghosh, D. et al. Learning to Reach Goals via Iterated Supervised Learning. In *International Conference on Learning Representations (ICLR)*, 2021. 4

[16] He, K. et al. Deep Residual Learning for Image Recognition. In *2016 IEEE/CVF CVPR*, 2016. 4

[17] Hirose, N. et al. Deep Visual MPC-Policy Learning for Navigation. *IEEE RA-L*, 4(4):3184–3191, October 2019. 4

[18] Hirose, N. et al. SACSoN: Scalable Autonomous Control for Social Navigation. *IEEE RA-L*, 9(1):49–56, January 2024. 4

[19] Hoeller, D. et al. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. *IEEE RA-L*, 6(3):5081–5088, July 2021. 1

[20] Hu, J. et al. FLaRe: Achieving Masterful and Adaptive Robot Policies with Large-Scale Reinforcement Learning Fine-Tuning. In *1st Workshop on X-Embodiment Robot Learning*, November 2024. 2

[21] Hu, Y. et al. Data Scaling Laws in Imitation Learning for Robotic Manipulation. In *1st Workshop on X-Embodiment Robot Learning*, November 2024. 1

[22] Höfer, S. et al. Sim2Real in Robotics and Automation: Applications and Challenges. *IEEE Transactions on Automation Science and Engineering*, 18(2):398–400, April 2021. 1, 2

[23] Kadian, A. et al. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? *IEEE RA-L*, 5(4):6670–6677, October 2020. 1, 2

[24] Kahn, G., Abbeel, P. and Levine, S. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. *IEEE RA-L*, 6(2):1312–1319, April 2021. 1, 2

[25] Karnan, H. et al. Socially CompliAnt Navigation Dataset (SCAND): A Large-Scale Dataset of Demonstrations for Social Navigation. *IEEE RA-L*, 7(4): 11807–11814, October 2022. 4

[26] Kaufmann, E. et al. Deep Drone Acrobatics. In *Robotics: Science and Systems XVI*, volume 16, July 2020. 2

[27] Kirillov, A. et al. Segment Anything. In *2023 IEEE/CVF ICCV*, 2023. 2

[28] Kulkarni, M. and Alexis, K. Reinforcement Learning for Collision-free Flight Exploiting Deep Collision Encoding. In *2024 IEEE ICRA*, May 2024. 1

[29] Loquercio, A. et al. Learning high-speed flight in the wild. *Science Robotics*, 6(59), October 2021. 1, 2

[30] Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. 2019. 4

[31] Majumdar, A. et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems*, volume 37, May 2024. 2, 3

[32] Masone, C. and Caputo, B. A Survey on Deep Visual Place Recognition. *IEEE Access*, 9:19516–19547, 2021. 2

[33] Meng, X. et al. Scaling Local Control to Large-Scale Topological Navigation. In *2020 IEEE ICRA*, May 2020. 2

[34] Mousavian, A. et al. Visual Representations for Semantic Target Driven Navigation. In *2019 IEEE ICRA*, May 2019. 1, 2

[35] Oquab, M. et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 2

[36] Radford, A. et al. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M. and Zhang, T., editors, *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. 2021. 2

[37] Ramakrishnan, S.K. et al. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1, December 2021. 4, 6

[38] Ramrakhya, R. et al. Habitat-Web: Learning Embodied Object-Search Strategies from Human Demonstrations at Scale. In *2022 IEEE/CVF CVPR*, June 2022. 1

[39] Ross, S., Gordon, G. and Bagnell, D. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. June 2011. 4

[40] Sadeghi, F. and Levine, S. CAD2RL: Real Single-Image Flight Without a Single Real Image. In *Robotics: Science and Systems XIII*, July 2017. 2

[41] Sadek, A. et al. An in-depth experimental study of sensor usage and visual reasoning of robots navigating in real environments. In *2022 ICRA*, May 2022. 1

[42] Sadek, A. et al. Multi-Object Navigation in real environments using hybrid policies. In *2023 IEEE ICRA*, May 2023. 2

[43] Savinov, N., Dosovitskiy, A. and Koltun, V. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[44] Savva, M. et al. Habitat: A Platform for Embodied AI Research. In *2019 IEEE/CVF ICCV (ICCV)*, Seoul, Korea (South), October 2019. 4

[45] Shah, D. et al. ViNG: Learning Open-World Navigation with Visual Goals. In *2021 IEEE ICRA*. 2021. 1

[46] Shah, D. et al. Rapid Exploration for Open-World Navigation with Latent Goal Models. In *Conference on Robot Learning*. January 2022. 4

[47] Shah, D. et al. GNM: A General Navigation Model to Drive Any Robot. In *2023 IEEE ICRA*, May 2023. 1, 2, 4

[48] Shah, D. et al. ViNT: A Large-Scale, Multi-Task Visual Navigation Backbone with Cross-Robot Generalization. In *Conference on Robot Learning*. August 2023. 1, 2, 3, 4, 5, 6

[49] Shang, J. et al. Theia: Distilling Diverse Vision Foundation Models for Robot Learning. In *Conference on Robot Learning*. September 2024. 2, 6

[50] Silwal, S. et al. What Do We Learn from a Large-Scale Study of Pre-Trained Visual Representations in Sim and Real Environments? In *2024 IEEE ICRA*, May 2024. 1, 2

[51] Spencer, J. et al. Feedback in Imitation Learning: The Three Regimes of Covariate Shift, February 2021. arXiv:2102.02872 [cs, stat]. 3, 5

[52] Sridhar, A. et al. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. In *2024 IEEE ICRA*, May 2024. 1, 2, 5

[53] Sun, X. et al. FGPrompt: Fine-grained Goal Prompting for Image-goal Navigation. *Advances in Neural Information Processing Systems*, 36:12054–12073, December 2023. 3

[54] Suomela, L. et al. PlaceNav: Topological Navigation through Place Recognition. In *2024 IEEE ICRA*, May 2024. 2, 4, 5

[55] Tan, M. and Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning (ICML)*. May 2019. 6

[56] Triest, S. et al. TartanDrive: A Large-Scale Dataset for Learning Off-Road Dynamics Models. In *2022 ICRA*, May 2022. 4

[57] Truong, J. et al. Rethinking Sim2Real: Lower Fidelity Simulation Leads to Higher Sim2Real Transfer in Navigation. In *Conference on Robot Learning*. March 2023. 1, 2, 4

[58] Truong, J. et al. IndoorSim-to-OutdoorReal: Learning to Navigate Outdoors Without Any Outdoor Experience. *IEEE RA-L*, 9(5):4798–4805, May 2024. 2

[59] Weinzaepfel, P. et al. CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In *Advances in Neural Information Processing Systems*, volume 34, October 2022. 3

[60] Yang, J.H. et al. Pushing the Limits of Cross-Embodiment Learning for Manipulation and Navigation. In *Robotics: Science and Systems XX*, volume 20, July 2024. 1

[61] Yang, L. et al. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In *2024 IEEE/CVF CVPR*, 2024. 2

[62] Zeng, K.H. et al. PoliFormer: Scaling On-Policy RL with Transformers Results in Masterful Navigators. In *Conference on Robot Learning*. September 2024. 2

[63] Zhai, X. et al. Sigmoid Loss for Language Image Pre-Training. In *2023 IEEE/CVF ICCV (ICCV)*, October 2023. 2

[64] Zhu, Y. et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE ICRA*, May 2017. 2