IG-Net: Image-Goal Network for Offline Visual Navigation on A Large-Scale Game Map

Pushi Zhang[1*], **Baiting Zhu**[2*], **Xin-Qiang Cai**[3*], **Li Zhao**[1],
Masashi Sugiyama[4,3] , **Jiang Bian**[1]
[1] Microsoft Research Asia, Beijing, China
[2] Stanford University, Stanford, United States
[3] The University of Tokyo, Tokyo, Japan
[4] RIKEN AIP, Tokyo, Japan

Abstract

Navigating vast and visually intricate gaming environments poses unique challenges, especially when agents are deprived of absolute positions and orientations during testing. This paper addresses the challenge of training agents in such environments using a limited set of offline navigation data and a more substantial set of offline position data. We introduce the *Image-Goal Network* (IG-Net), an innovative solution tailored for these challenges. IG-Net is designed as an image-goal-conditioned navigation agent, which is trained end-to-end, directly outputting actions based on inputs without intermediary mapping steps. Furthermore, IG-Net harnesses position prediction, path prediction and distance prediction to bolster representation learning to encode spatial map information implicitly, an aspect overlooked in prior works. Results demonstrate IG-Net's potential in navigating large-scale gaming environments, providing both advancements in the field and tools for the broader research community.

1 Introduction

Visual navigation, the act of autonomously traversing and understanding environments based on visual cues, has been at the forefront of robotics and artificial intelligence research (Shah et al., 2021, 2023b; Kwon et al., 2021). The ability to navigate is a fundamental skill for agents, making it applicable in a wide range of scenarios, from virtual gaming environments to real-world robotic applications. The challenge, however, lies in the complexity and variability of these environments, especially when the scale is vast and the available data is limited.

In this work, we consider the ShooterGame environment with realistic visual dynamics[1], which spans 10421.87 m^2 across multiple floors, representing a scale approximately 50-100 times larger than preceding navigational environments. Furthermore, we focus on the setting that the agent only has access to limited offline navigation data but can use some random unlabeled data (without action or continuity) to enhance the model training. In contrast, the testing phase further restricts the agent's access to solely the current and goal RGB observations. The observational data is limited to a 90 FoV that aligns with the human player's view, posing considerable challenges compared to the conventional 360 FoV camera observations (Al-Halah et al., 2022).

To mitigate these challenges, we propose the *Image-Goal Network* (IG-Net), an end-to-end solution specifically designed for large-scale visual navigation tasks. This network amalgamates visual

[*] Equal contribution. Work done at Microsoft Research Asia. Contact: pushizhang@microsoft.com.

[1] Figure 1 in the Appendix provides an illustration of the ShooterGame.

NeurIPS 2023 Workshop on Robot Learning: Pretraining, Fine-Tuning, and Generalization with Large Scale Models, New Orleans, United States of America

and positional information to guide the agent towards its goal effectively. Besides, since explicitly building a map for navigating such a large-scale environment is challenging, we incorporate spatial information, representing the positional information of each image implicitly for a better representation. Experiments of IG-Net on the large-scale ShooterGame map demonstrate significant advancements in navigation ability compared to baselines, opening avenues for further research and development in autonomous navigation in large-scale, complex environments.

2 Related Works

Image-goal visual navigation within large-scale maps, particularly when devoid of absolute positions during testing and online interactions during training, poses a profound challenge addressed by numerous research endeavors. Advancements in topology-based methodologies have been noteworthy. The Topological Semantic Graph Memory (Kim et al., 2022), utilizing depth cameras, constructs graphs based on images or objects and avoids reliance on positional information, employing a cross-graph mixer for updates. Similarly, Visual Graph Memory (Kwon et al., 2021) leverages landmark-based topological representations for zero-shot navigation in novel environments, and the Neural Topological SLAM (Chaplot et al., 2020) updates graphs through nodes representing 360-degree panoramic views based on agent observations. Mod-IIN (Krantz et al., 2023) uses an exploration method that maintains a top-down 2D map with depth and positional inputs for efficient visual navigation. ViNG (Shah et al., 2021) predicts steps and accessibility to a target while generating waypoints, constructing trees, and planning paths via a weighted Dijkstra algorithm. ViNT (Shah et al., 2023c) builds topological graphs for long-horizon navigation, while RNR-Map (Kwon et al., 2023) builds renderable neural radiance map for long-horizon navigation. Compared to these works, our work develops multiple auxiliary tasks for learning an implicit map, without the requirement of modeling the environment with explicit topological graphs or maps.

Some existing works, including Visual Graph Memory (Kim et al., 2022) and ViNT (Shah et al., 2023c), have incorporated pretraining methods for visual navigation as a component. Compared to these works, we focus on long-horizon navigation setting in the pretraining, and study more types of pretraining objectives that utilizes positional information for visual navigation.

Some pretraining methods were proposed to facilitate robot learning across a variety of environments. The visual representation R3M (Nair et al., 2022) demonstrates the potential of data-efficient learning for downstream robotic manipulation tasks using pre-trained visual representations on diverse human video data. PACT (Bonatti et al., 2022) introduces a generative transformer-based architecture that builds robot-specific representations from robot data in a self-supervised fashion, evidencing enhanced performance in tasks such as safe navigation. Also, Majumdar et al. (2023) conducts an extensive empirical study focusing on the design of an artificial visual cortex, aimed at enabling an artificial agent to convert camera input into actions. Our method shares the idea of representation learning for embodied agents, but we explore how a specific type of supervision: positional information, benefits the learning of visual navigation agents.

3 Problem Setting

In this study, we tackle the intricate problem of visual navigation within the significantly expansive ShooterGame environment. This environment features a 10422 m^2 map with multiple levels, making it approximately 50-100 times larger than previous navigation environments in Table 3. Additionally, we use a 90 FoV camera view *without* depth as opposed to a panoramic view shown in Figure 2. Other ShooterGame details can be found in Appendix A.

Offline Dataset. We collect navigation data to train our models. At each step, the agent is given the current RGB observation $\boldsymbol{o}_t$ and the target $\boldsymbol{o}_{tar}$. A human-expert action $\boldsymbol{a}_t \in \{\textit{forward}, \textit{turn left}, \textit{turn right}\}$ is executed which leads to next observation. The trajectory ends when agent navigates to target. Global positions and rotations $\boldsymbol{p}_t = (x_t, y_t, z_t, \theta_t)$ are also collected but only used for auxiliary tasks and visualizations. Each trajectory in our dataset is represented as:

$$\tau = \{\boldsymbol{o}_{tar}, \boldsymbol{o}_0, \boldsymbol{p}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{o}_T, \boldsymbol{p}_T, \boldsymbol{a}_T\}, \tag{1}$$

where T denotes trajectory length. A total of $N = 200$ trajectories are used for training with an average length of 55 steps. Other details can be found in Appendix F.1

We collect additional position data that includes the positions-images of the map. We uniformly sample a set of $M = 2000$ points in the map, where each point is represented as the observation-position-rotation pair $\{\boldsymbol{o}^j, \boldsymbol{p}^j | 1 \leq j \leq M\}$. Note that position data are unavailable during testing.

4 Proposed Method: IG-Net

4.1 Foundation Principles

Given the extensive scale of the map, constructing a model explicitly based on the map, such as topological methods (Kim et al., 2022), is not feasible. Accordingly, IG-Net is proposed with the following distinct properties to navigate proficiently within such constrained settings. Additional discussions on IG-Net can be found in Appendix B.1 to B.3.

End-to-End Training: Distinct from methodologies that construct a comprehensive map or graph of the environment as a separate step, IG-Net adopts end-to-end training. This allows IG-Net to directly interpret inputs and output actions, bypassing intermediary mapping processes.

Enhanced Representation Learning: IG-Net utilizes the position data and navigation information prediction to refine representation learning, a domain relatively untouched in preceding studies. It employs a variety of auxiliary tasks detailed below to enhance the agent's spatial understanding.

4.2 Training Tasks

To optimize IG-Net, we devise a conglomerate of training objectives, each catering to different aspects of navigation. These tasks ensure the coherent learning of representations and navigational strategies, which are crucial for effective navigation in complex environments. We here discuss the high-level motivation of each task and present other details such as loss functions in Appendix C.

Relative Position: Given one current image $\boldsymbol{o}_1$ and goal image $\boldsymbol{o}_2$, we use IG-Net to predict the difference in position and orientation between the pair. This allows IG-Net to learn to inherit spatial representation given camera views on the map.

Absolute Position: We use IG-Net to predict the absolute position and rotations given a camera view. For the first two losses, we train on both offline navigation data and position data.

Navigation Distance: Given one current image and one goal image, IG-Net learns to predict the total distance that takes the agent to navigate from the first state to the second state. Specifically, we want IG-Net to capture state connectivity through training on this loss.

Navigation Path: Given one current image and one goal image, we train IG-Net to construct a spatial navigation path between them. We locally predict the trajectory for the next 5 steps and globally predict the relative positions at 5 equally-spaced steps from current step t to T.

Low-level Action: Given one current image and one goal image, we use an additional action loss for training IG-Net to generate the current navigation action.

5 Experiment

5.1 Experiment Setting

We evaluate IG-Net in three levels of difficulties based on the initial Euclidean distance between agent and goal. We run 50 episodes under each setting with a maximum of 200 steps per episode. Success is marked by the agent locating within a fixed range of the goal, regardless of its orientation.

IG-Net is compared against VGM (Kwon et al., 2021), ViNT(Shah et al., 2023c), NoMaDSridhar et al. (2023), and GNM Shah et al. (2023a). All algorithms are trained with the same dataset as described in Sec 3. Due to limited simulation speed of ShooterGame(2 fps), we disabled RL training on all algorithms. Notice NoMaD, ViNT, and GNM's performance might suffer from mismatched action space.

5.2 Evaluation Metrics

Three metrics are used in this paper: success rate (SR), success weighted by path length (SPL), and distance decrement rate (DDR).

SPL measures the efficiency of navigation and is defined as $SPL = \frac{1}{N} \sum_{i=1}^{N} S_i \frac{d_i}{max(d_i,p_i)}$ where S_i equals 1 if navigation is successful and 0 otherwise. N is the total number of evaluation episodes, d_i is shortest distance to target approximated by Euclidean D_i, and p_i is the actual trajectory length.

DDR measures the closest distance achieved between the agent and the target towards to and can be written as $DDR = \frac{D-d_{min}}{D}$ where D is the initial Euclidean distance to the goal and d_{min} is the minimum Euclidean distance throughout the trajectory.

5.3 Results

Results are presented in Table 2. Under the easiest setting, IG-Net's success rate outperforms other baselines by a margin of 69% (from 0.32 to 0.54) and is 71% more efficient in SPL (from 0.14 to 0.24). More remarkably, IG-Net achieves a reasonable SR of 0.24 to 0.26 under more challenging settings whereas all other baselines almost completely fail with SR consistently below 0.1. Case study and additional analysis can be found in E

Moreover, results show that training IG-Net with auxiliary tasks significantly improves performance in both success rate and navigation efficiency. Therefore, we conclude the learning objectives proposed in Section 4.2 help IG-Net to establish an implicit and transferable understanding of the map.

Difficulty	$15m < D < 40m$			$40m < D < 80m$			$D > 80m$		
Metric	SR	SPL	DDR	SR	SPL	DDR	SR	SPL	DDR
VGM	0.32	0.14	0.49	0.04	0.02	0.26	0.00	0.00	0.27
NoMaD-EMA	0.08	0.05	0.27	0.06	0.04	0.26	0.00	0.00	0.23
NoMaD	0.10	0.06	0.31	0.02	0.01	0.20	0.00	0.00	0.16
ViNT	0.08	0.07	0.17	0.02	0.02	0.16	0.00	0.00	0.15
GNM	0.04	0.04	0.13	0.04	0.03	0.15	0.02	0.02	0.14
IG-Net	**0.54**	**0.24**	**0.75**	**0.26**	**0.17**	**0.65**	**0.24**	**0.15**	**0.75**
IG-Net (no auxiliary)	0.18	0.09	0.36	0.14	0.08	0.42	0.00	0.00	0.44

Table 1: IG-Net experiment results. SR: success rate. SPL: success-weighted by path length. DDR: distance decrement rate. D: initial distance to goal.

5.4 Generalization to Novel Maps

We validate IG-Net's ability to navigate in novel maps on Gibson. The model is trained on 396 different Gibson environments then evaluated on 72 environments from the training set and 14 unseen ones. Both training and testing on Gibson follow easy setting with initial distance $1.5m \leq D \leq 3m$. Where success is remarked by agent calling a *stop* action within $1m$ range to goal.

Setting	Train			Eval (unseen)		
Metric	SR	SPL	DDR	SR	SPL	DDR
IG-Net	**0.54**	**0.47**	**0.45**	**0.58**	**0.51**	**0.42**
IG-Net (no auxiliary)	0.00	0.00	0.32	0.00	0.00	0.32

Table 2: IG-Net experiment results on Gibson.

6 Conclusion

In this study, we tackled visual navigation in expansive gaming environments with the introduction of the Image-Goal Network (IG-Net). IG-Net is a testament to the synergy of cutting-edge deep learning and specialized navigation strategies, emphasizing image-goal-conditioned behavior and the implicit encoding of spatial map information, a facet underexplored in preceding works. The network's proven adaptability and robustness in the expansive ShooterGame environment underscore its potential in navigating large-scale, visually rich domains using solely offline, image-centric data. The significant advancements of IG-Net are not confined to enhancing visual navigation but extend to enriching representation learning, providing invaluable insights for ongoing and future investigations in both virtual and real-world autonomous navigation applications. The foundational principles of IG-Net are poised to influence the development of more sophisticated navigation agents.

References

Ziad Al-Halah, Santhosh K. Ramakrishnan, and Kristen Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 17010–17020. IEEE, 2022.

Rogerio Bonatti, Sai Vemprala, Shuang Ma, Felipe Frujeri, Shuhang Chen, and Ashish Kapoor. PACT: perception-action causal transformer for autoregressive robotics pre-training. *CoRR*, abs/2209.11133, 2022.

Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, pp. 667–676. IEEE Computer Society, 2017.

Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological SLAM for visual navigation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 12872–12881. Computer Vision Foundation / IEEE, 2020.

Onno Eberhard, Jakob Hollenstein, Cristina Pinneri, and Georg Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Nuri Kim, Obin Kwon, Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwai Oh. Topological semantic graph memory for image-goal navigation. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski (eds.), *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pp. 393–402. PMLR, 2022.

Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. *arXiv preprint arXiv:2304.01192*, 2023.

Obin Kwon, Nuri Kim, Yunho Choi, Hwiyeon Yoo, Jeongho Park, and Songhwai Oh. Visual graph memory with unsupervised representation for visual navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 15870–15879. IEEE, 2021.

Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9099–9108, 2023.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? *CoRR*, abs/2303.18240, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski (eds.), *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 2022.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas A. Funkhouser, and Vladlen Koltun. MINOS: multimodal indoor simulator for navigation in complex environments. *CoRR*, abs/1712.03931, 2017.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pp. 13215–13222. IEEE, 2021.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A General Navigation Model to Drive Any Robot. In *International Conference on Robotics and Automation (ICRA)*, 2023a. URL `https://arxiv.org/abs/2210.03370`.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *CoRR*, abs/2306.14846, 2023b.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023c.

Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 190–198. IEEE Computer Society, 2017.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. *arXiv pre-print*, 2023. URL `https://arxiv.org/abs/2310.07896`.

Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 9068–9079. Computer Vision Foundation / IEEE Computer Society, 2018.

Appendix

A Environment Details

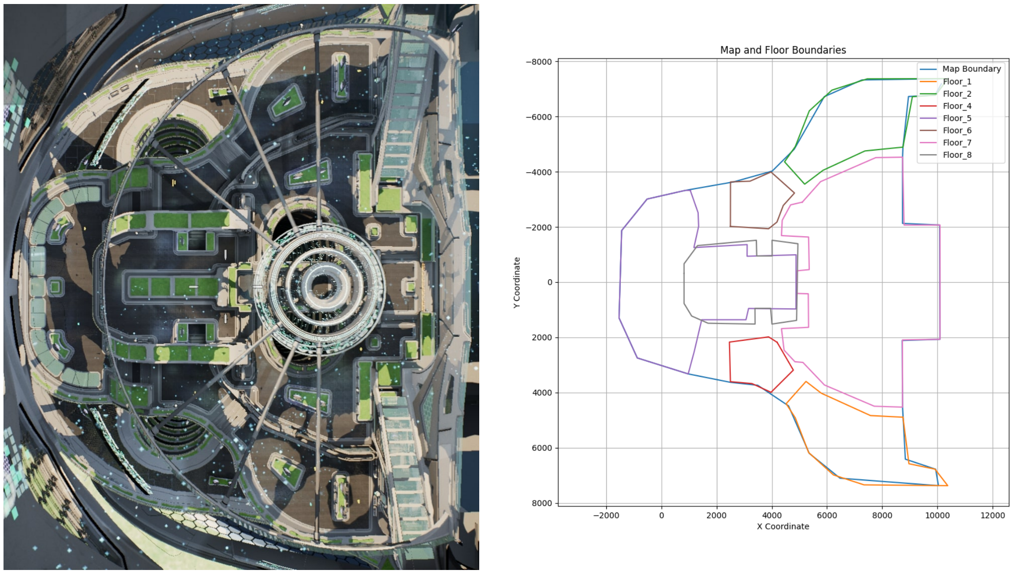
ShooterGame is a quintessential representation of a PC multiplayer first-person shooter by Unreal Engine 4, providing a robust framework that includes diverse weapon implementations, game modes, and a simplistic front-end menu system, with observations further constrained to 90-degree camera views [2]. This restriction augments the challenge compared to preceding 360-degree camera observations as in Figure 2. From the figure we can also observe that the distant craft, clouds, sunlight, and even walls change dynamically over time, resulting in different observations of the same position and angle at different moments in time, which renders the navigation within this environment a complex endeavor. The lack of depth information also poses unique implementations and challenges for navigation tasks.

B Model Details

B.1 IG-Net Architectural Design

IG-Net integrates a transformer-based structure, specifically tailored for navigation tasks, with a forward process described as follows:

[2] A public video demo: `https://www.youtube.com/watch?v=xdS6asajHAQ`



(a) A bird view of ShooterGame

(b) A sketch of the ShooterGame

Figure 1: A bird view and a sketch of usable space of ShooterGame.

Environment	Gibson	SUNCG	Matterport3D	ShooterGame
Coverage of One Map (m^2)	368.88	127.13	517.78	10421.87
Dynamic Objects	✗	✗	✗	✓
Pure RGB without depth	✗	✗	✗	✓
No Panoramic 360 camera view	✗	✗	✗	✓

Table 3: Comparison of navigation environments including Gibson (Xia et al., 2018), SUNCG (Song et al., 2017), Matterport3D (Chang et al., 2017), and MINOS (Savva et al., 2017). The coverage of one task of ShooterGame is calculated using a polygon area calculation method applied to the entire map. It is 50-100 times bigger than previous ones. The scale is converted from the game engine's base units.

1. **Image Encoding:** A pretrained Masked Auto-Encoder (MAE) is employed for encoding the current and goal images independently, ensuring a rich representation of visual information.
2. **Embedding Concatenation:** The encoding embeddings procured from the first step are concatenated to form a unified representation, encompassing both current state and goal state information.
3. **Positional and Action Decoding:** Utilizing position, path, distance, and action decoders, the network predicts corresponding positional information and navigational actions, leveraging the concatenated embeddings.

B.2 IG-Net Training and Inference

During training, IG-Net is exposed to a plethora of offline navigation data, enriched with positional and visual information. The network learns to intertwine visual and spatial cues to formulate robust navigational policies. In inference, the network, confined to current observational and goal images, generates actions to navigate the agent proficiently toward the predefined goal, overcoming the constraints imposed by limited observational data and expansive environments.

B.3 IG-Net Parameters

The hyperpameters of training IG-Net is provided as follows:

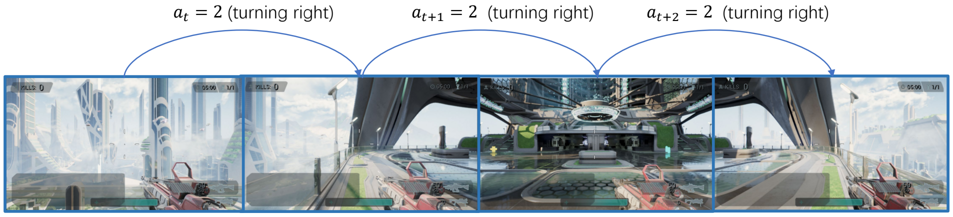

a_t = 2 (turning right)
a_{t+1} = 2 (turning right)
a_{t+2} = 2 (turning right)

Figure 2: Four image observations from one fixed position with different view angles. Each observation only contains a 90-degree camera view, which can be more challenging than previous 360-degree-view environments.

Param Name	IG-Net
Learning Rate	$5e^{-5}$
Batch Size	16
N (Number of Navigation Trajectories)	200
M (Number of Position Data)	2000
Visual Dim	$224 \times 224 \times 3$
Visual Backbone	MAE
Max Epoch	100
Action Dim	3
Path prediction length	5
Loss weights for each auxiliary loss	1.0
Loss weights action loss	1.0

B.4 VGM Parameters

We mostly follow the default VGM training procedure by keeping the architecture of the model the same as in the original publication. To train under ShooterGame, we cut panoramic observations to 90 FoV and mask the depth input. To overcome the issue of having sparse nodes during training, we tune the th parameter to different values to loosen node generation criteria. Finally, we train for a maximum of 250 epochs and choose the model checkpoint when validation loss achieves the lowest.

Param Name	Default	Tuned
Learning Rate	$1e^{-4}$	-
Batch Size	4	-
Th	0.75	0.85
Visual Dim	$64 \times 64 \times 3$	-
Visual Backbone	ResNet-18	-
Max Epoch	250	-
Action Dim	3	-

C Loss Function Details

Relative Position Prediction. The relative position prediction design allows IG-Net to learn to inherit spatial representation given camera views on the map. Given any two states represented by $(\boldsymbol{o}_1, \boldsymbol{p}_1)$ and $(\boldsymbol{o}_2, \boldsymbol{p}_2)$, we compute the relative position and orientation of these two states as:

$$\mathrm{relative}(\boldsymbol{p}_2, \boldsymbol{p}_1) = \left((x_2 - x_1, \quad y_2 - y_1, \quad z_2 - z_1)\, R(-\theta_1)^T, \theta_2 - \theta_1 \right), \tag{2}$$

where $R(\theta)$ is the rotation matrix for angle θ. Qualitatively, $\mathrm{relative}(\boldsymbol{p}_2, \boldsymbol{p}_1)$ reflects the position and rotation of $\boldsymbol{o}_2$ in the egocentric coordinates of $\boldsymbol{o}_1$. Given a pair of images, IG-Net is able to predict the relative position of the images, and the following loss function is used for relative position prediction in IG-Net:

$$L^{\mathrm{relative}}(\boldsymbol{o}_1, \boldsymbol{p}_1, \boldsymbol{o}_2, \boldsymbol{p}_2) = L^{\mathrm{pos\_angle}}(f_\theta^{\mathrm{relative}}(\boldsymbol{o}_1, \boldsymbol{o}_2), \mathrm{relative}(\boldsymbol{p}_2, \boldsymbol{p}_1)), \tag{3}$$

where

$$
\begin{aligned}
&L^{\text{pos\_angle}}((x_1, y_1, z_1, \theta_1), (x_2, y_2, z_2, \theta_2)) \\
&= \|(x_2 - x_1, y_2 - y_1, z_2 - z_1, \cos(\theta_2) - \cos(\theta_1), \sin(\theta_2) - \sin(\theta_1))\|_2^2
\end{aligned}
$$

evaluate how the predicted relative positions and rotations are close to the ground truth relative positions and rotations.

One advantage of relative position is that any data with position information can be leveraged. We use a mixture of position offline data and navigation offline data for training the relative position prediction, detailed later in this section.

Absolute Position Prediction. We additionally use IG-Net to predict the absolute position and rotations given a camera view, serving as an additional auxiliary loss for IG-Net. Given one state represented by $(\boldsymbol{o}_1, \boldsymbol{p}_1)$, the following loss function is used for training IG-Net is given by:

$$
L^{\text{absolute\_pos}}(\boldsymbol{o}_1, \boldsymbol{p}_1) = L^{\text{pos\_angle}}(f_\theta^{\text{absolute\_pos}}(\boldsymbol{o}_1), \boldsymbol{p}_1). \tag{4}
$$

We also use a mixture of offline navigation data and position data for training IG-Net.

Navigation distance prediction. For the navigation distance prediction task, IG-Net is given a pair of states represented by image observations, and learns to predict the total distance that takes the agent to navigate from the first state to the second state. When the loss is optimized, the network captures the connectivity between different states in the map. Given a trajectory $\tau = (\boldsymbol{o}_{tar}, \boldsymbol{p}_{tar}, \boldsymbol{o}_0, \boldsymbol{p}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{o}_T, \boldsymbol{p}_T, \boldsymbol{a}_T)$ in the offline navigation dataset, we let $\boldsymbol{o}_{T+1} = \boldsymbol{o}_{tar}$ and $\boldsymbol{p}_{T+1} = \boldsymbol{p}_{tar}$ define the navigation distance between $\boldsymbol{o}_i, \boldsymbol{o}_j, i \leq j$ as follows:

$$
\text{nav\_distance}(\boldsymbol{o}_i, \boldsymbol{o}_j, \tau) = \sum_{k=i}^{j-1} \|(x_k - x_{k+1}, y_k - y_{k+1}, z_k - z_{k+1})\|_2^2. \tag{5}
$$

Given a pair of states in the offline navigation dataset, IG-Net predicts the navigation distance between them. The loss function for training IG-Net is given by:

$$
L^{\text{nav\_distance}}(\boldsymbol{o}_i, \boldsymbol{o}_j, \tau) = \left[f_\theta^{\text{nav\_distance}}(\boldsymbol{o}_i, \boldsymbol{o}_j) - \text{nav\_distance}(\boldsymbol{o}_i, \boldsymbol{o}_j, \tau)\right]^2 \tag{6}
$$

Navigation path prediction. Given a pair of states represented by image observations, IG-Net learns to construct the spatial navigation path between them, serving as a path-planning auxiliary loss for IG-Net. For the local path prediction in IG-Net, the predicted path is the $N_{path} = 5$ next consecutive steps in the navigation trajectory; for the global path prediction in IG-net, the predicted path is the $N_{path} = 5$ intermediate relative positions, where the intermediate points are equally spaced in time from current time t to the total path length T.

Formally, we define the local and global timesteps as

$$
S^{\text{local}}(t, \tau) = (\min(t+1, T), \min(t+2, T), \ldots, \min(t + N_{path}, T)), \tag{7}
$$

$$
S^{\text{global}}(t, \tau) = \left(t + \lfloor \frac{T-t}{N_{path}+1} \rfloor, t + \lfloor \frac{2(T-t)}{N_{path}+1} \rfloor, \cdots, t + \lfloor \frac{N_{path}(T-t)}{N_{path}+1} \rfloor \right). \tag{8}
$$

We define the local and global path at timestep t in the trajectory τ as

$$
\text{local\_path}(\boldsymbol{o}_t, \tau) = \left(\text{relative}(\boldsymbol{p}_{S^{\text{local}}(t,\tau)_1}, \boldsymbol{p}_t), \cdots, \text{relative}(\boldsymbol{p}_{S^{\text{local}}(t,\tau)_{N_{path}}}, \boldsymbol{p}_t)\right), \tag{9}
$$

$$
\text{global\_path}(\boldsymbol{o}_t, \tau) = \left(\text{relative}(\boldsymbol{p}_{S^{\text{global}}(t,\tau)_1}, \boldsymbol{p}_t), \cdots, \text{relative}(\boldsymbol{p}_{S^{\text{global}}(t,\tau)_{N_{path}}}, \boldsymbol{p}_t)\right). \tag{10}
$$

Finally, the training loss on local and global paths for IG-Net is defined as:

$$
L^{\text{local\_path}}(\boldsymbol{o}_t, \tau) = \sum_{k=1}^{N_{path}} \left[L^{\text{pos\_angle}}(f_\theta^{\text{local\_path}}(\boldsymbol{o}_t, \boldsymbol{o}_{tar})_k, \text{local\_path}(\boldsymbol{o}_t, \tau)_k)\right], \tag{11}
$$

$$
L^{\text{global\_path}}(\boldsymbol{o}_t, \tau) = \sum_{k=1}^{N_{path}} \left[L^{\text{pos\_angle}}(f_\theta^{\text{global\_path}}(\boldsymbol{o}_t, \boldsymbol{o}_{tar})_k, \text{global\_path}(\boldsymbol{o}_t, \tau)_k)\right]. \tag{12}
$$

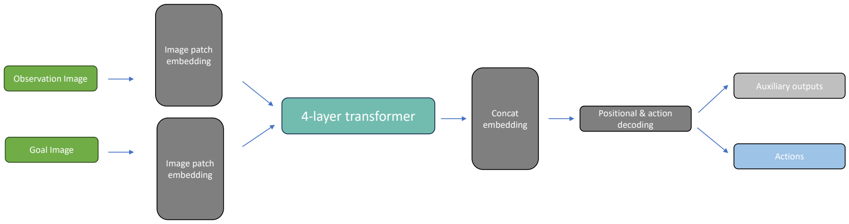


Figure 3: IG-Net architecture illustration, in which auxiliary outputs denote predictions on map positions & orientations; ego positions & orientations; local & global paths.

Low-level Action Loss. Besides all the above auxiliary loss, we use an additional action loss for training IG-Net to generate navigation actions. Given one current image and one goal image, we train IG-Net to predict the current action. The action prediction head of IG-Net is trained with behavior cloning loss:

$$L^{\text{action}}(\boldsymbol{o_t}, \tau) = \text{cross\_entropy}(f^{\text{action}}(\boldsymbol{o}_t, \boldsymbol{o}_{tar}), a_t) \tag{13}$$

Training Loss. We add all the auxiliary loss and action prediction loss as a single loss function to train IG-Net. We use $w = 1.0$ for each loss term in our experiment.

Sampling in position and navigation dataset for training IG-Net. All the position prediction losses are trained with both position and navigation datasets. In contrast, navigation distance, path prediction loss and action loss rely solely on the navigation dataset. In our experiment, we sample the position dataset with $p_{pos} = 0.4$ probability and the navigation dataset with $1 - p_{pos} = 0.6$ probability. When sampled on the position dataset, the navigation distance and path prediction loss are masked in training. Our approach enables leveraging both the position and navigation datasets for training different auxiliary tasks without losing data efficiency.

D Architectural Design

IG-Net integrates a transformer-based structure, specifically tailored for navigation tasks, with a forward process described as follows:

1. **Image Encoding:** A pretrained Masked Auto-Encoder (MAE) is employed for encoding the current and goal images independently, ensuring a rich representation of visual information.
2. **Embedding Concatenation:** The encoding embeddings procured from the first step are concatenated to form a unified representation, encompassing both current state and goal state information.
3. **Positional and Action Decoding:** Utilizing position, path, distance, and action decoders, the network predicts corresponding positional information and navigational actions, leveraging the concatenated embeddings.

An illustration of the architecture of IG-Net is shown in Figure 3.

E Case Study

E.1 Visualization of Navigation Path of IG-Net

To demonstrate IG-Net's proficiency in visual navigation, especially with purely visual inputs in complex environments such as ShooterGame, we present case studies depicting the navigation paths executed by IG-Net during the evaluation phase, as illustrated in Figure 4. From its initial position, IG-Net successfully executes its planning paths and executes low-level actions seamlessly, navigating through stairs and corridors while avoiding collisions with obstacles. These observations are consistent across various evaluation episodes, showcasing IG-Net's capability to navigate accurately towards the goal image and execute precise low-level navigational maneuvers to follow the correct path.

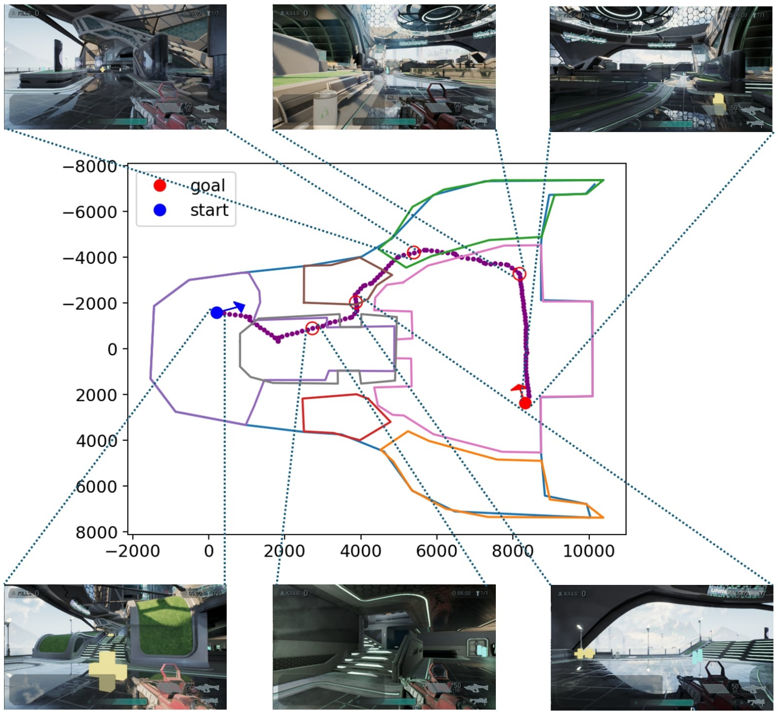


Figure 4: Illustration of IG-Net's navigation path in ShooterGame during evaluation. The bottom-left figure represents the agent's starting position, and the top-right figure displays the goal image, serving as input to IG-Net. Purple dots trace the path navigated by IG-Net, and red dots represent key frames in the navigation, with corresponding images visualized.

E.2 Robustness of IG-Net

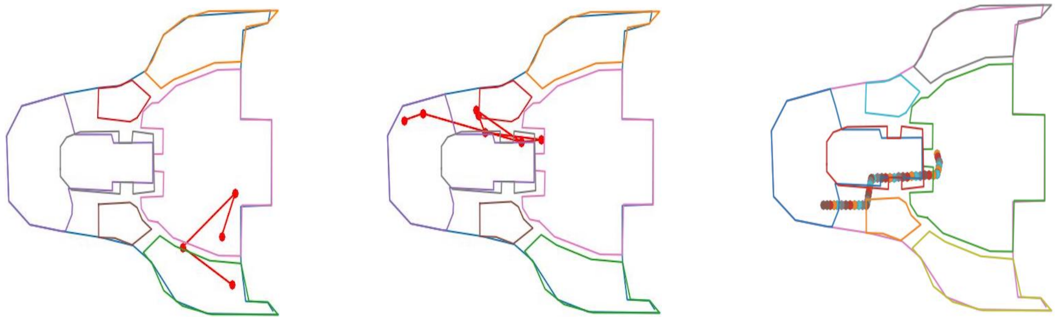
To assess IG-Net's robustness, we conduct a case study introducing Gaussian noises, denoted as n, to the positions. We normalize of the position to zero-mean and unit-variance, and add a noise on all position training signals with standard derivation of n. Table 4 reveals that IG-Net maintains substantial performance even amidst high noise levels. Intriguingly, noise appears to enhance IG-Net's performance in challenging tasks ($D > 8000$), a phenomenon akin to utilizing noise to augment agents' exploration capability in RL scenarios (Eberhard et al., 2023; Plappert et al., 2018; Fortunato et al., 2018). This unexpected benefit opens up promising avenues for future enhancements to IG-Net's performance.

Difficulty	$1500 < D < 4000$			$4000 < D < 8000$			$D > 8000$		
Metric	SR	SPL	DDR	SR	SPL	DDR	SR	SPL	DDR
IG-Net	**0.54**	**0.24**	**0.75**	**0.26**	**0.17**	**0.65**	0.24	0.15	**0.75**
IG-Net ($n = 0.1$)	0.26	0.13	0.47	0.22	0.14	0.61	0.16	0.11	0.64
IG-Net ($n = 0.2$)	0.42	0.18	0.58	0.16	0.09	0.58	**0.30**	**0.20**	0.74
IG-Net ($n = 0.4$)	0.26	0.12	0.52	0.18	0.09	0.61	0.20	0.12	0.70

Table 4: Performance of IG-Net under different noise levels.

E.3 Why VGM Fails

VGM, along with several other methodologies (Kim et al., 2022), strives to represent environments using nodes and vertices, relying solely on visual information. Our findings suggest that in expansive gaming environments like ShooterGame, graph construction is prone to failure and necessitates meticulous hyperparameter tuning (refer to B.4). Moreover, the nodes in VGM often encompass only a minor section of the large-scale map, hindering the algorithm from utilizing prior map information to facilitate new navigation tasks.



(a) Too sparse ($th = 0.75$) (b) Too sparse ($th = 0.85$) (c) Too dense ($th = 1.25$)

Figure 5: Illustration of failed VGM node construction under varying parameters.

Furthermore, VGM nodes often only cover a minor section of the large-scale map, which prevents the algorithm from leveraging prior map information to guide new navigation tasks.

E.4 Ablation study

In this section, we explore the auxiliary tasks' contribution to representation learning and the subsequent enhancement of IG-Net's navigation capabilities. The results are detailed in Table 5. It is evident that the absence of various auxiliary tasks leads to performance degradation to varying degrees. The IG-Net (no aux) variant, lacking all auxiliary losses, exhibits the most considerable performance decline. These results conclusively show that the designed auxiliary tasks significantly enrich IG-Net's representation and, consequently, elevate its navigation performance.

Difficulty	$1500 < D < 4000$			$4000 < D < 8000$			$D > 8000$		
Metric	SR	SPL	DDR	SR	SPL	DDR	SR	SPL	DDR
IG-Net	**0.54**	**0.24**	**0.75**	0.26	**0.17**	**0.65**	0.24	0.15	**0.75**
IG-Net (no position)	0.30	0.14	0.43	**0.28**	0.14	0.59	0.12	0.07	0.55
IG-Net (no path and dist)	0.38	0.17	0.58	0.26	0.14	0.62	**0.30**	**0.20**	0.66
IG-Net (no auxiliary)	0.18	0.09	0.36	0.14	0.08	0.42	0.00	0.00	0.44

Table 5: Ablation study on the impact of auxiliary losses.

F Dataset Details

F.1 Training Dataset

All training navigation trajectories are collected by human experts. There are a total of 200 trajectories and each takes less than 2 minutes to collect. Human experts have prior experience with the game environment and are given additional information such as goal location on the map and distance to the goal to facilitate efficient collection. We here provide some descriptive details of the dataset.

Stats Name	Val
Num of Trajs	200
Total Nav Steps	10617
Avg Nav Steps	55.87
Max Traj Len	130
Min Traj Len	9
Avg D_0 to Goal	5800
Max D_0 to Goal	13386
Min D_0 to Goal	809

F.2 Evaluation Dataset

Evaluation are carried out in 3 difficulties: easy, medium, and hard distinguished by the initial Euclidean distance to goal D_0. The specific ranges are: $1500 < D_0^{easy} < 4000$, $4000 < D_0^{medium} < 8000$, and $8000 < D_0^{hard}$. Notice success is marked by the agent's Euclidean distance to the goal is within 800. We here provide some descriptive details of the dataset

Stats Name	Easy	Medium	Hard
Avg D_0	2673	5888	9404