Differentiable $\mathrm{E}(2)$-Equivariant Graph Planning for Navigation [‡]

Linfeng Zhao[1*], Hongyu Li[1,2*], Taşkın Padır[1], Huaizu Jiang[1†], and Lawson L.S. Wong[1†]

Abstract— **Learning for robot navigation presents a critical and challenging task. The scarcity and costliness of real-world datasets necessitate efficient learning approaches. In this letter, we exploit Euclidean symmetry in planning for 2D navigation, which originates from Euclidean transformations between reference frames and enables parameter sharing. To address the challenges of unstructured environments, we formulate the navigation problem as planning on a geometric graph and develop an equivariant message passing network to perform value iteration. Furthermore, to handle multi-camera input, we propose a learnable equivariant layer to lift features to a desired space. We conduct comprehensive evaluations across five diverse tasks encompassing structured and unstructured environments, along with maps of known and unknown. Our experiments confirm the substantial benefits on training efficiency, stability, and generalization.**

I. Introduction

Navigation is a fundamental capability of mobile robots. Traditional navigation approaches, such as A* [2], focus on finding shortest-distance collision-free paths to a provided goal location in a pre-built occupancy map or known costmap. Recently, learning-based approaches to robot navigation have been proposed [3–7], which are particularly useful when the costs or goals are not explicitly provided and need to be learned from data. For example, in visual navigation, the cost of navigation between locations may depend on high-dimensional visual features, and the goal may also need to be visually identified (*e.g.*, “find a mug”). As another example, in imitation learning, users may provide information about their preferred navigation policy implicitly via demonstrations, and the costs or optical actions need to be learned using features from the robot’s state-action space.

While the aforementioned learning-based approaches exhibit remarkable capability in handling high-dimensional observations, they typically require a considerable amount of data and intensive training [3, 4]. Furthermore, these methods lack guarantees regarding generalization capabilities. In this work, we investigate the potential benefits of Euclidean symmetry in navigation tasks. It stems from Euclidean transformations among reference frames, enabling parameter sharing, enhancing efficiency, and improving generalizability. The utilization of symmetry in navigation within the *grid world domain* is explored in the earlier study by Zhao et al. [8] (left of Fig. 1). They introduce the equivariant version of the value iteration network (VIN) [9] under *discrete translations, rotations, and reflections*, along with a differentiable navigation planner. Their work showcases notable improvements compared to baseline approaches [9, 10]. However, they only focused on navigation in discrete 2D grids, which limits its applicability to robot navigation.

In our work, we introduce an equivariant learning-based navigation approach that operates on graphs in continuous space and considers symmetry with respect to an infinitely larger continuous group – the Euclidean group $\mathrm{E}(2)$ (right of Fig. 1). Specifically, we use *geometric graphs* (or spatial graphs) [11], where nodes in our graph correspond to states (and their features) arbitrarily located in 2D space. This eliminates the confinement to a grid, enabling the environment to remain non-discretized and permitting variable resolution. This also helps when the robot’s motion deviates from grid-like patterns. Moreover, our approach accounts for continuous rotational symmetry, enhancing learning efficiency compared to discrete symmetry like Dihedral group D_4.

However, to exploit Euclidean symmetry in graph-based navigation, we need to solve two major challenges. First, previous work on 2D grids exploited the grid nature of their problem and used standard 2D symmetric convolution, which is no longer applicable in our case. Instead, we derive a new $\mathrm{E}(2)$-equivariant message-passing version of VIN and validate that it satisfies our notion of symmetry. Second, to capture symmetry in visual inputs/features, previous work relied on a very specific setup. As illustrated in Fig. 1, the agent was assumed to have four cameras, each situated 90° apart, exactly matching the D_4 symmetry being considered, such that a group transformation (rotation) can be implemented as a permutation to the four images. Extending this approach directly to $\mathrm{E}(2)$ would technically require an infinite number of cameras (or at least an infinite-resolution panoramic camera). We lift this restriction by introducing a learnable equivariant layer that can take images from a camera array conforming to a subgroup of $\mathrm{E}(2)$ (such as D_8) and lift their features to become $\mathrm{E}(2)$-equivariant.

We empirically demonstrate the effectiveness of our approach on various navigation environments, including 2D grid, 2D geometric graphs, and Miniworld visual navigation [12] on both grid and graph. Moreover, in demonstrating its potential suitability for semantic goals and real-world environments, we provide a proof-of-concept experiment on semantic navigation tasks in the Habitat simulator [13]. Among these studies, we observe a consistent improvement in learning efficiency and stability. Overall, our study provides insight into the application of equivariance in navigation and the challenges. Our contributions are three-fold:

$*$ equal contribution. † equal advising.

‡ A full version of this paper was published at RA-L [1].

[1] Northeastern University, Boston, MA. {zhao.linf, t.padir, h.jiang, l.wong}@northeastern.edu.

[2] Brown University, Providence, RI. hongyu@brown.edu. This work was partially done when Hongyu Li was at Northeastern University.

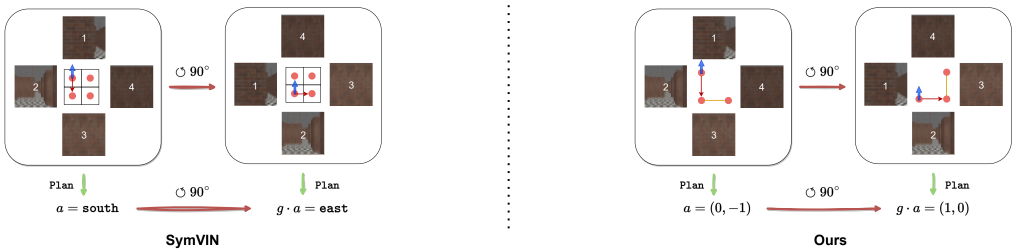

Fig. 1: **Illustration of rotation equivariance.** We provide a side-by-side comparison with SymVIN [8]. We use the blue arrow to show the orientation of the robot. Rotating the robot ↻ 90° is equivalent to rotating the world frame ↻ 90°. When camera views are cyclically permuted, action output (red arrow) is transformed by a rotation matrix. The state space of SymVIN (left) is confined to the grid, and it only produces discrete actions. Our approach acts on continuous 2D space and produces $\mathbb{R}^2$ actions.

- We study the equivariance properties of 2D navigation and identify the two challenges.
- To address the challenges, we (1) derive the geometric message passing (MP) version of value iteration on geometric graphs and (2) propose using a learnable equivariant layer that converts multi-camera images to desired feature space, respectively.
- We demonstrate the empirical performance of navigation on Grid World (2D grid), Graph World (2D geometric graphs), and Miniworld visual navigation on both grid and graph. We provide proof-of-concept results on semantic navigation in Habitat simulator.

II. Background and Problem Formulation: Navigation as Geometric Graphs

Definition. We approach navigation as a 2D continuous path planning problem, building upon the 2D discrete grid version introduced in [8, 9], while extending it to the utilization of the *geometric graph* $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ in 2D Euclidean space $\mathbb{R}^2$. In navigation tasks, the agent observes a state $\boldsymbol{s}_t \in \mathcal{S}$ at each step, and the action is to move on the 2D plane $\boldsymbol{a}_t \in \mathcal{A} = \mathbb{R}^2$. State $\boldsymbol{s}_t$ can be a 2D position in $\mathbb{R}^2$ or egocentric panoramic images in $\mathbb{R}^{K \times H \times W}$ (where K denotes the number of images of resolution $H \times W$)[1]. To convert the task into a geometric graph, each node $v_i \in \mathcal{V}$ corresponds to a state $\boldsymbol{s} \in \mathcal{S}$ and is associated with a *node feature* $\boldsymbol{h}_i$ (such as images) and has a position $\boldsymbol{x}_i \in \mathbb{R}^2$. It is also possible to use *edge features*[2]. In this paper, we focus on addressing the global planning problem: given a navigation task (state $\boldsymbol{s}$ and target $\boldsymbol{w}$) as a feature field/map M, we output action field $\Pi = \texttt{Plan}_\theta(M)$.

Geometric Structure. The navigation problem can be defined as a MDP, and an inherent geometric structure emerges: it can be conceptualized as a geometric graph (defined above) situated within *a 2D Euclidean space*. Specifically, this graph can be transformed through 2D Euclidean isometric symmetries, without impacting the optimal solution of the MDP [8, 14]. The set of all such transformations in 2D is called *Euclidean group* $\mathrm{E}(2)$, which can be uniquely decomposed into translation part $\mathbb{R}^2$ and rotation/reflection part $\mathrm{O}(2)$, denoted as semi-direct product $\rtimes$: $\mathrm{E}(2) = \mathbb{R}^2 \rtimes \mathrm{O}(2)$ [15, 16]. We only require the node features $\boldsymbol{h}$ (and edge features) are transformable by a subgroup $G \leq \mathrm{E}(2)$. Following the notation in [16], we denote the rotation/reflection symmetry part as compact symmetry group $G \leq \mathrm{GL}(2)$, because translation group is *not compact* and many useful theorems do not hold. In our implementation, translation equivariance is achieved by using *relative position*. For any subgroups G of rotation/reflection, its equivariance needs *group convolution* [15] or *steerable convolution* [17].

Value Iteration and Symmetry. When symmetry appears in an MDP, the value and policy functions are equivariant [8, 14]. Abstractly, we can write value iteration ($\texttt{VI}$) as iteratively applying the Bellman operator $\mathcal{T} : V_t \mapsto V_{t+1}$:

$$\begin{aligned} Q_t(\boldsymbol{s},\boldsymbol{a}) &:= R(\boldsymbol{s},\boldsymbol{a}) + \int_{\mathbb{R}^2} d\boldsymbol{s}' P(\boldsymbol{s}' \mid \boldsymbol{s},\boldsymbol{a}) V(\boldsymbol{s}'), \\ V_{t+1}(\boldsymbol{s}) &= \max_{\boldsymbol{a}} Q_t(\boldsymbol{s},\boldsymbol{a}), \end{aligned} \tag{1}$$

where the input and output of the Bellman operator are both value function $V : \mathcal{S} \to \mathbb{R}$. Specifically, $\boldsymbol{s} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}, R(\boldsymbol{s},\boldsymbol{a}), P(\boldsymbol{s}' \mid \boldsymbol{s},\boldsymbol{a})$ represent the states, actions, rewards, and transitions, respectively. In VIN, $\texttt{VI}(M) = \mathcal{T}_M^k [V_0]$ is executed k times, which takes an initial value V_0 and the map M (with a goal) as input

$$g \cdot \texttt{VI}(M) \equiv g \cdot \mathcal{T}_M^k [V_0] = \mathcal{T}_M^k [g \cdot V_0] \equiv \texttt{VI}(g \cdot M). \tag{2}$$

Zhao et al. [8] explore the equivariance for a 2D grid case. We extend it to geometric graph: $\mathcal{T}$ is performed on a graph, which is implemented using message passing.

Symmetry Transformations. In this paragraph, we unify the concepts presented in the preceding two paragraphs to demonstrate the implementation of equivariance constraints, which establish equivalence between transformed and original input/output [15, 17, 18]. Under the group transformation g, a (left) *regular* representation L_g transforms a feature map with c_{out}-dimensional vector (vector field) $f : X \to \mathbb{R}^{c_{\text{out}}}$ as [11, 17, 18]:

$$\left[L_g f\right](x) = \left[f \circ g^{-1}\right](x) = \rho_{\text{out}}(g) \cdot f\left(g^{-1} x\right), \tag{3}$$

where ρ_{out} is the G-representation associated with output $\mathbb{R}^{c_{\text{out}}}$. For example, for *action* feature map $\Pi : \mathbb{R}^2 \to \mathbb{R}^2$ (i.e., every position $\boldsymbol{x} \in \mathbb{R}^2$ is associated with a relative 2D movement), rotating the vector needs a 2×2 rotation matrix.

[1] We omit image RGB channel for notation simplicity.

[2] Similarly, each edge $e_{ij} \in \mathcal{E}$ corresponds to a state-action transition $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$ and has an *edge feature* $\in \mathbb{R}^{c_e}$ (such as distance or movement cost).

III. Methodology: Equivariant Message Passing for Value Iteration

Following the spirit of VIN, we build a geometric message passing network and extend it to learning value iteration on geometric graphs: $\Pi = \texttt{Plan}_\theta(M)$. Given that the input feature map M and the resulting action map Π are both amenable to transformation within the same group, we enforce equivariance constraints throughout the MP network (shown in Fig. 1): $g \cdot \Pi = \texttt{Plan}_\theta(g \cdot M)$.

A. Message Passing Value Iteration Networks (MP-VIN)

Regarding the value iteration process, our MP-VIN is analogous to the original VIN formulation. However, what sets our approach apart is the improvement brought about by the inclusion of the geometric graph using an equivariant message passing layer (discussed in the next section). There are two advantages of using graph format: (1) cover the environment with irregular graphs to achieve variable resolution, and (2) output continuous actions.

B. $\mathrm{O}(2)$-Equivariant Message Passing Layer: Equivariant Value Iteration on Graph

Discretization to Graph. We could employ standard 2D convolution on regular grids for value iteration, as seen in VIN. However, irregular graphs render grids unsuitable. In the prior study of Niu et al. [19], an earlier iteration of graph convolution was employed. However, it exhibited equivariance only with respect to $\mathbb{R}^2$ translations, and it did not encompass considerations for rotation or reflection symmetries ($\mathrm{O}(2)$). Here, we derive from first principles using the original continuous form of value iteration.

The integral term in VI can be written as a mapping Φ

$$\boldsymbol{h}'(\boldsymbol{x}) = \Phi[h](\boldsymbol{x}) = \int_{\mathbb{R}^2} \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')\boldsymbol{h}(\boldsymbol{x}'), \tag{4}$$

where $\boldsymbol{K} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^{c_\text{out} \times c_\text{in}}$ is the kernal function [3]. $\boldsymbol{h} : \mathbb{R}^2 \to \mathbb{R}^{c_\text{in}}$ and $\boldsymbol{h}' : \mathbb{R}^2 \to \mathbb{R}^{c_\text{out}}$ are input and output feature map. If translation equivariance is desired, the kernel can be further simplified from two-argument to one-argument case, and the mapping is *convolution* [8, 20]. The continuous steerable convolution $\star$ is defined (via cross-correlation) by [16–18]:

$$\boldsymbol{h}'(\boldsymbol{x}) = [\boldsymbol{K} \star \boldsymbol{h}](\boldsymbol{x}) = \int_{\mathbb{R}^2} \boldsymbol{K}(\boldsymbol{x}' - \boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}'), \tag{5}$$

where $\boldsymbol{K} : \mathbb{R}^2 \to \mathbb{R}^{c_\text{out} \times c_\text{in}}$ is a (steerable) kernel.

If we sample nodes in $\mathbb{R}^2$ and construct edges by transition $\mathcal{S} \times \mathcal{A}$, the continuous convolution on $\mathbb{R}^2$ can be discretized, which is similar to strategy of PointConv [21] [4]. We use *nonlinear message passing* to replace linear convolution. We use two MLPs for computing messages ($\texttt{propagate}_\theta$) and updating node features ($\texttt{update}_\theta$), and has form

$$\boldsymbol{m}_{ij} = \texttt{propagate}_\theta\left(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{x}_i, \boldsymbol{x}_j\right),$$
$$\boldsymbol{h}'_i = \texttt{update}_\theta\left(\boldsymbol{h}_i, \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij}\right). \tag{6}$$

Implementation of Equivariance. We implement $\mathrm{E}(2)$-equivariant message passing on the graph that is equivariant under two parts:

Translation $\mathbb{R}^2$. In the plannar convolution on 2D grid, it is known to be equivariant to translation because it only relies on relative position between two cells as input and never takes absolute coordinates. Analogously, we use relative position between nodes $\boldsymbol{x}_i - \boldsymbol{x}_j$ as input to the message passing function [21]:

$$\boldsymbol{m}_{ij} = \texttt{propagate}_\theta\left(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{x}_i - \boldsymbol{x}_j\right). \tag{7}$$

It is a direct generalization of translation-equivariant 2D convolution that relies only on relative positions or local coordinates (shown in Eq. 5), allowing generalization to larger maps.

Rotation and Reflection $\mathrm{O}(2)$. We use steerable equivariant network to implement $\mathrm{O}(2)$-equivariance [15–17, 20, 21]. The $\mathrm{O}(2)$ group is compact and thus its representations are decomposable into *irreducible representations* [16, 18], thus convolutions can be performed in Fourier domain and more efficient. We use it to build equivariant MLPs of $\texttt{propagate}$ and $\texttt{update}$ (effectively 1×1 convolution). The kernel $\boldsymbol{K}$ of G-steerable convolution needs to satisfy constraint [16, 18], where G can be any (discrete) subgroup of $\mathrm{O}(2)$:

$$\boldsymbol{K}(gx) = \rho_\text{out}(g) \circ \boldsymbol{K}(x) \circ \rho_\text{in}(g)^{-1} \quad \forall g \in G, x \in \mathbb{R}^2, \tag{8}$$

where ρ_in and ρ_out stand for representations of the layer's input and output, respectively. This kernel constraint guarantees that the layer is G-equivariant: $\boldsymbol{K}(gx)\rho_\text{in}(g) = \rho_\text{out}(g) \circ \boldsymbol{K}(x)$. We refer the readers to Weiler and Cesa [16], Cohen and Welling [17] for more details.

C. C_K-Equivariant Lifting Layer: Processing Camera Array

In the previous section, we extend from discrete symmetry in SymVIN to continuous symmetry, such as continuous rotations $\mathrm{SO}(2)$. Injecting such equivariance into the *entire* network requires us to know how to **continuously** rotate sensory input by $g \in SO(2)$. This can be naturally achieved by two types of observations: (1) 360° point cloud input from a LiDAR (naturally continuous) or (2) 360° cylindrical camera. However, (1) may not seamlessly incorporate semantic information from RGB images, and (2) is hard to obtain and process. Thus, we need to relax this requirement of the $\mathrm{SO}(2)$-transformable input modality. As a solution, we introduce a learnable layer $\texttt{lift}$ that can map camera images from different views to a $\mathrm{SO}(2)$-transformable feature. This enhances our ability to exploit symmetry in the planning process.

[3] Note that the kernel $\boldsymbol{K}$ here is different from the notation K we use to represent the number of images.

[4] Brandstetter et al. [21] discuss other strategy for 3D steerable messsage passing, which expands the feature maps to spherical harmonics. Analogously, it is possible to expand the features to cyclic harmonics.

TABLE I: Averaged test success rate (%) with standard deviation. The best result is **bolded**. The second-best result is underlined.

Method	Grid World		Graph World		Miniworld	
	15×15	27×27	128 nodes	256 nodes	Grid	Graph
VIN [9]	78.51±1.81	50.15±3.94	18.75±1.95	20.09±6.87	57.14±8.92	18.90±2.87
SymVIN [8]	**95.85**±5.02	**93.73**±7.33	24.40±2.11	27.53±4.73	**91.67**±2.58	27.98±4.34
MP-VIN (No Sym)	87.07±4.53	55.99±39.56	63.10±17.26	54.76±3.29	–	–
MP-VIN: D_8	87.19±2.78	38.53±16.17	52.38±5.02	32.89±3.47	–	–
MP-VIN: $\mathbb{R}^2$	90.81±1.02	72.45±31.94	70.24±2.69	58.33±5.93	79.76±24.06	96.58±2.46
MP-VIN: $\mathbb{R}^2 \rtimes D_8$	91.50±1.04	84.52±6.04	**72.17**±5.08	**61.90**±5.33	90.89±1.63	**96.96**±1.00

Although the output is $\mathrm{SO}(2)$-transformable, the left side is only C_4-transformable, so the layer `lift` can only be *restricted* to be C_4-equivariant. The restriction from $G = \mathrm{SO}(2)$ to subgroup $H = C_4$ is called *restricted representation.* This layer is a special kind of equivariant induction layer [22]. It can lift features on a subgroup $H \leq G$ to a group G and is H-equivariant. Intuitively, it needs to satisfy the equivariance constraint only for $\circlearrowleft 90^\circ \in C_4$, which is a subgroup $C_4 \leq \mathrm{SO}(2)$:

$$\texttt{lift}(\circlearrowleft 90^\circ \cdot \texttt{images}) = \circlearrowleft 90^\circ \cdot \texttt{features}, \tag{9}$$

where we assume 4 images and output $\mathrm{SO}(2)$ features, while it can be any group such that C_4 is its subgroup.

IV. Experiments

We evaluate our proposed approach MP-VIN and baselines on four different tasks. Among these tasks, we perform point goal navigation under different environments: known structured environments (`Grid World`), known unstructured environments (`Graph World`), unknown structured environments (`Miniworld`), and unknown unstructured environments (`Miniworld-Graph`).

Methods. We experiment four variants of our methods, with or without translation ($\mathbb{R}^2$) or rotation/reflection (using $G = D_8 \leq \mathrm{O}(2)$) equivariance: No-Sym, D_8, $\mathbb{R}^2$, and $\mathbb{R}^2 \rtimes D_8$. We use two grid-based methods: VIN [9] and SymVIN [8] (with D_4-equivariance).

A. Planning on known maps: Grid World

Results. In terms of absolute performance gain, by adding D_8 symmetry to MP-VIN with $\mathbb{R}^2$, we obtain another 0.69% and 12.07% success rate on the 15×15 and 27×27 mazes, respectively. However, it is still outperformed by SymVIN, which uses steerable 2D convolution to process the input. It is reasonable as it directly uses the regular grid structure, while our graph version can handle unstructured grpahs and is more expressive, while we apply both of them on grid maps. When the map size increases, MP-VIN with $\mathbb{R}^2 \rtimes D_8$ symmetry demonstrates the second-least performance degradation, showing better generalization to larger maps.

B. Planning on known graphs: Graph World

Results. The graphs generally do not have regular structure, i.e. four neighbors only in four directions. Thus, all methods encounter performance degradation, while grid-based methods struggle more in such unstructure graphs.

C. Mapping and planning under unknown maps: Miniworld

Results. Since this task is based on a 15×15 grid using visual observations, the experiment results are similar to the Grid World. MP-VIN with $\mathbb{R}^2 \rtimes D_8$ symmetry demonstrates higher learning efficiency than MP-VIN with only $\mathbb{R}^2$ symmetry and VIN. However, every method faces a performance drop due to the mapping uncertainty. We observe that the performance gap of MP-VIN with $\mathbb{R}^2 \rtimes D_8$ symmetry (0.61%) is lower than that of SymVIN (4.18%). Therefore, the performance gap between MP-VIN with $\mathbb{R}^2 \rtimes D_8$ symmetry and SymVIN becomes narrower.

D. Mapping and planning under unknown graphs: Miniworld-Graph

Results. Similar to the Miniworld experiment on grid representation, we observe the MP-VIN with $\mathbb{R}^2 \rtimes D_8$ symmetry has higher learning efficiency than MP-VIN with only $\mathbb{R}^2$ symmetry. Grid-based approaches suffer in this task since it is hard for CNN to process the expressive unstructured environment, also indicated in the previous `Graph World` experiment. In both Miniworld experiments, we observe that MP-VIN with $\mathbb{R}^2 \rtimes D_8$ symmetry has a much smoother learning curve and lower variance than MP-VIN with only $\mathbb{R}^2$ symmetry. This indicates that by adding $\mathbb{R}^2$ symmetry, we could further optimize the network with better stability.

V. Conclusion

In this paper, we explored the applicability of exploiting Euclidean symmetry within the context of a navigation planner. We contributed a novel equivariant differentiable planner. The effectiveness of the proposed approach is extensively assessed across four distinct tasks involving structured and unstructured environments, with known and unknown maps. The empirical findings demonstrate a significant enhancement in learning efficiency when Euclidean symmetry is integrated into 2D navigation planning. Furthermore, the results indicate that leveraging Euclidean symmetry yields more stable optimization and yields superior overall performance. In the future, we hope to extend our work to navigation task that has higher dimension, such as semantic navigation [3–5].

[4] One solution for D_4 group is to use *quotient* representations, but it not generally applicable for higher-degree rotations such as D_8 or infinitesimal rotations $\mathrm{SO}(2)$.

References

[1] L. Zhao, H. Li *et al.*, "E(2)-equivariant graph planning for navigation," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, 2024.

[2] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968, conference Name: IEEE Transactions on Systems Science and Cybernetics.

[3] D. S. Chaplot, D. P. Gandhi *et al.*, "Object Goal Navigation using Goal-Oriented Semantic Exploration," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 4247–4258.

[4] M. Chang, A. Gupta, and S. Gupta, "Semantic Visual Navigation by Watching YouTube Videos," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 4283–4294.

[5] Y. Liang, B. Chen, and S. Song, "SSCNav: Confidence-Aware Semantic Scene Completion for Visual Semantic Navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 13 194–13 200, iSSN: 2577-087X.

[6] N. U. Akmandor, H. Li *et al.*, "Deep Reinforcement Learning based Robot Navigation in Dynamic Environments using Occupancy Values of Motion Primitives," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 11 687–11 694, iSSN: 2153-0866.

[7] A. Brohan, N. Brown *et al.*, "RT-1: Robotics Transformer for Real-World Control at Scale," Dec. 2022, arXiv:2212.06817 [cs].

[8] L. Zhao, X. Zhu *et al.*, "Integrating Symmetry into Differentiable Planning," in *ICLR 2023*. ICLR, Jun. 2022, arXiv:2206.03674 [cs] type: article.

[9] A. Tamar, Y. WU *et al.*, "Value Iteration Networks," in *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016.

[10] L. Lee, E. Parisotto *et al.*, "Gated Path Planning Networks," *arXiv:1806.06408 [cs, stat]*, Jun. 2018, arXiv: 1806.06408.

[11] M. M. Bronstein, J. Bruna *et al.*, "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges," *arXiv:2104.13478 [cs, stat]*, Apr. 2021, arXiv: 2104.13478.

[12] M. Chevalier-Boisvert, "Miniworld: Minimalistic 3d environment for rl & robotics research," https://github.com/maximecb/gym-miniworld, 2018.

[13] M. Savva, A. Kadian *et al.*, "Habitat: A Platform for Embodied AI Research," 2019, pp. 9339–9347.

[14] E. van der Pol, D. E. Worrall *et al.*, "MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning," *arXiv:2006.16908 [cs, stat]*, Jun. 2020, arXiv: 2006.16908.

[15] T. S. Cohen and M. Welling, "Group Equivariant Convolutional Networks," *arXiv:1602.07576 [cs, stat]*, Jun. 2016, arXiv: 1602.07576.

[16] M. Weiler and G. Cesa, "General $E(2)$-Equivariant Steerable CNNs," *arXiv:1911.08251 [cs, eess]*, Apr. 2021, arXiv: 1911.08251.

[17] T. S. Cohen and M. Welling, "Steerable CNNs," Nov. 2016.

[18] L. Lang and M. Weiler, "A Wigner-Eckart Theorem for Group Equivariant Convolution Kernels," Sep. 2020.

[19] S. Niu, S. Chen *et al.*, "Generalized Value Iteration Networks: Life Beyond Lattices," *arXiv:1706.02416 [cs]*, Oct. 2017, arXiv: 1706.02416.

[20] T. Cohen, M. Geiger, and M. Weiler, "A General Theory of Equivariant CNNs on Homogeneous Spaces," *arXiv:1811.02017 [cs, stat]*, Jan. 2020, arXiv: 1811.02017.

[21] J. Brandstetter, R. Hesselink *et al.*, "Geometric and Physical Quantities Improve E(3) Equivariant Message Passing," *arXiv:2110.02905 [cs, stat]*, Dec. 2021, arXiv: 2110.02905.

[22] O. Howell, D. Klee *et al.*, "Equivariant single view pose prediction via induced and restricted representations," *arXiv preprint arXiv: 2307.03704*, 2023.