Learning to Explore (L2E): Deep Reinforcement Learning-based Autonomous Exploration for Household Robot

Zuxin Liu[1*,2], Mohit Deshpande[1], Xuewei (Tony) Qi[1], Ding Zhao[2], Rajasimman Madhivanan[1], Arnie Sen[1]
[1]Amazon Lab126, [2]Carnegie Mellon University

***Abstract*—We study the autonomous exploration task in indoor environments for the mobile ground robot. We propose a three-stage exploration strategy: viewpoint generation, viewpoint scoring, and viewpoint selection, to make the algorithm agnostic to the robot's planning and control modules. In particular, we propose the Learning to Explore (L2E) framework, which formulates the scoring and selection stages as a learning problem that could be solved by imitation learning (IL) and deep reinforcement learning (DRL). We use IL to pretrain the exploration policy and an off-policy DRL method to fine-tune it, improving the sample efficiency and accelerating the training process. We benchmark both the heuristic-based viewpoint scoring and selection methods and the proposed DRL method under the same exploration framework in realistic diverse indoor environments, and the results show that the L2E method can achieve $4\% \sim 22\%$ minimum improvement when compared with baseline exploration approaches.**

I. INTRODUCTION

Autonomous exploration is crucial for robots operating in unknown environments, with applications spanning household [1, 2, 3] to rescue scenarios [4, 5, 6]. It serves to guide robots in perceiving and mapping their surroundings, an essential step for subsequent tasks like localization and navigation [7, 8]. Our focus lies on indoor autonomous exploration, where the robot navigates a 2D plane to construct an occupancy grid map of the environment [9, 10]. Consumer-grade robots, like Amazon Astro [11], often equipped with cheaper, narrower field-of-view (FoV) sensors, pose unique challenges, which our approach can address.

It is still a challenging problem to find an efficient, scalable, and generalizable exploration framework [12]. The efficiency is determined by the new information gained from a fixed budget of resources, such as the exploration area within a certain amount of time. It is one of the most important metrics to evaluate the performance of an exploration planner since better efficiency indicates less exploration time and energy consumption. The scalability is reflected by the algorithm complexity regarding the explored map size. Poor scalability may restrict its deployment on the real robot in large environments. Finally, a generalizable planner should be easily adapted to a new robotic platform or new environments.

Traditional rule-based methods for indoor robot exploration are abundant in literature [9, 3]. They employ heuristic-based cost functions to determine the next navigation goal, but such heuristics may not generalize well and their greedy nature can hamper long-term planning [12]. Furthermore, their scalability can falter in larger environments.

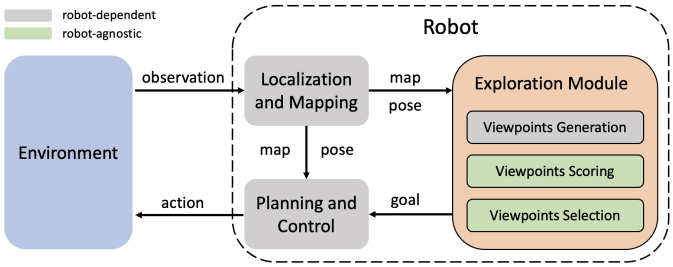


Fig. 1: Exploration framework.

Deep reinforcement learning (DRL) has shown promise in exploration, teaching robots to interact with the environment [13, 14]. These approaches offer scalability and environment adaptability, but have difficulties in comprehensive environment coverage and require extensive training data [15, 16]. Direct utilization of robot movement in the action space limits transferability across robotic platforms [17, 18].

We propose the exploration framework shown in Fig. 1 to address these challenges. The localization and mapping module, the planning and control module, and the viewpoints generation module are rule-based and robot-dependent, while the viewpoints scoring and selection modules are robot-agnostic because the inputs and outputs can share the same representation across different platforms. We propose the Learning to Explore (L2E) method, which includes a pretraining phase with imitation learning, and a fine-tuning phase where DRL is utilized to further improve the performance. Our contributions include:

- Proposing the L2E method, which combines rule-based and learning-based techniques for efficient, scalable, and generalizable indoor exploration.
- Framing autonomous exploration as a learning problem with a safe, feasible action space and model architecture, accelerated by IL pretraining and refined through DRL fine-tuning for optimal performance.
- Evaluating our approach against rule-based baselines in varied indoor environments using a narrow-FoV robot, showing significant improvements over baselines, particularly in complex scenarios.

II. METHOD

As shown in Fig. 1, the learning to explore (L2E) method consists of the viewpoints generation phase, the viewpoints scoring phase, and the viewpoints selection phase. We propose a rule-based method to generate candidate viewpoints that are guaranteed to be navigable. The downstream scoring and selection modules are robot-agnostic.

A. Viewpoints generation

Our viewpoint generation module starts by detecting frontiers, the boundaries between known and unknown spaces, using a wavefront detector ($\texttt{WFD}(P, M)$) [19]. Rather than waiting for complete frontier detection, we generate viewpoints concurrently with frontier discovery.

While clustering the frontier F, we incrementally compute and cache the midpoint $F.\texttt{midpoint}$ to efficiently constrain the search space, as this typically offers the most information gain. A Breadth-First Search from the midpoint is then initiated, calculating a frontier viewpoint that expands outwards towards known space across navigable cells. This search proceeds until a valid cell is identified, based on the following criteria:

- The cell is sufficiently distanced from obstacles and the frontier itself, ensuring we avoid obstacles and impasses.
- The cell maintains a line-of-sight to the frontier, ensuring an increment in the explored area post-navigation.

Upon locating a viewpoint v that fulfills these criteria, it's added to the viewpoint set $\mathcal{V}$, and the viewpoint for the subsequent frontier is computed. This module's operation on the high-level occupancy grid map and validation of line-of-sight to the frontier allows generalization to different sensor configurations and ranges, provided they generate the same occupancy grid map and can compute a line-of-sight to the frontier.

B. State space representation

Our approach formulates exploration as a learning problem using state representation easily digestible by neural networks. We adopt the occupancy grid, common in indoor robot state representation [20, 21]. It employs rasterized image representation akin to self-driving tasks [22, 23, 24, 25, 26]. This uses a fixed-size RGB image, encompassing all necessary task information. Fig. 2a shows various cells: white for free space, black for unknown, teal for obstacles, and purple for frontiers. Robot and viewpoints are depicted with red and blue circles respectively. We supplement the rasterized image with a robot feature vector and viewpoints' feature vectors. The compact and informative state representation allows us to facilitate the model training procedure.

C. Action space and reward function design

Different from previous learning-based methods that directly output the robot's action or predict the next viewpoint [17, 18, 16, 10], we adopt a discrete action space that includes all the candidate viewpoints from the generation phase. The exploration planner selects a viewpoint from the pre-generated ones as the next navigation goal at each decision time.

We argue that the proposed action space choice has four major benefits: 1) the output of the learning exploration policy is always safe and feasible, i.e., the robot will not collide with obstacles or navigate to invalid positions; 2) the performance of the learned exploration strategy is lower-bounded, 3) the learned exploration strategy is robot-agnostic, i.e., it could be easily transferred to another robot platform with different sensor configurations or navigation modules; 4) the training efficiency could be greatly improved, i.e., the learning agent can use fewer interaction samples and less training time to converge when compared to previous DRL methods that use continuous action spaces [15]. We enjoy the first three benefits because the viewpoints are validated to be navigable in the viewpoint generation phase by considering the robot configurations, therefore, any choices from the learning policy should be feasible and can obtain new information. The last benefit comes from the compressed action space from the entire exploration area to a fixed number of candidate viewpoints, which avoids additional training to figure out feasible and informative actions.

We use the difference of the explored area between two decision time steps as the reward function, which is straightforward. The reward signal indicates the new information gained by selecting the viewpoint.

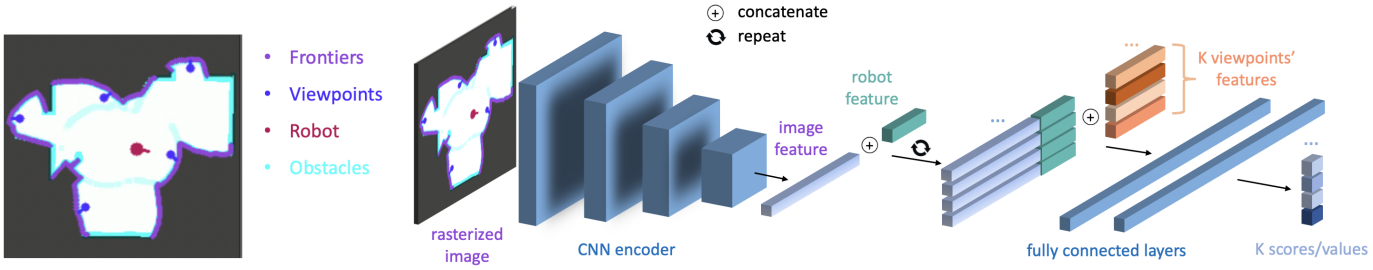
D. Model architecture

The proposed model architecture for our state and action spaces is depicted in Fig. 2b. The rasterized image, resized to $3 \times 224 \times 224$, is processed by a Convolutional Neural Network (CNN) encoder with multi-stage pooling to derive a condensed image feature vector. This vector represents the global information of the entire exploration map. The CNN backbone utilizes the EfficientNet-B0 model [27].

The model must manage variable numbers of viewpoint features and maintain output invariance irrespective of viewpoint feature input order. We address this by borrowing the scoring net concept from trajectory prediction [22, 26], which can accommodate fluctuating viewpoint quantities. The world feature vector, a concatenation of the image and robot features, encodes the exploration task's global information. This vector is then repeated K times and combined with each of the K viewpoint feature vectors. Here, K is the count of candidate viewpoints, each representing local information. Subsequently, the K feature vectors are fed through 2×512 hidden layers with ReLU activation to generate K scores, each linked to a corresponding viewpoint. Notably, this design treats the K dimension as a batch input to the network, resulting in order-invariant outputs capable of handling varying viewpoints. The policy head's scores are normalized using softmax to determine the probability of selecting each viewpoint.

E. Model training

The policy training has two stages: 1) using imitation learning (IL), particularly Behavior Cloning (BC) [28] in this work, to initialize the CNN encoder and 2) using DRL for fine-tuning. In the IL pretraining phase, we adopt the nearest frontier



(a) Rasterized image

(b) Model architecture

Fig. 2: Overview of the rasterized image representation and the model architecture.

(NF) planner that we will introduce in Sec. III-A as the expert policy to collect exploration demonstrations, i.e., sequences of state-action pairs. Then we train the policy network based on the behavior cloning loss function [28]. After IL training, we use the pretrained encoder for the DRL agent and randomly initialize the fully connected layers for the policy and value function heads. We only use the pretrained encoder to prevent the converged IL policy from being trapped to a local optimum.

We use the Soft-Actor-Critic (SAC) [29, 30] method to fine-tune the policy model in the second stage. Since SAC is off-policy, i.e., reusing the data in the replay buffer, it has better sample efficiency than previous on-policy DRL-based autonomous exploration methods [15, 16].

III. EXPERIMENTS

A. Baselines

We compare the proposed L2E approach with four rule-based exploration methods that are commonly used in the literature [15, 31, 16]. To fairly compare them, we implement all the baselines under the same exploration framework as shown in Fig. 1, where the viewpoints generation stage is the same while the scoring and selection stages are different. The decision frequency is 1Hz for all methods. Denote $\mathcal{V}$ as the set of viewpoints. We detail each baseline as follows.

Nearest Frontier (NF). The NF planner scores each candidate viewpoint v based on the navigation distance from the robot $\mathcal{D}(v)$, which is computed by the A* path planning algorithm, and then greedily selects the closest one as the next navigation goal: $v^* = \arg\max_{v\in\mathcal{V}}(-\mathcal{D}(v))$.

Information Gain (IG). The IG planner scores each candidate viewpoint based on the associated frontier cluster's size $\mathcal{I}(v)$, i.e., the number of corresponding frontier points. It greedily selects the one with the highest information gain as the next navigation goal: $v^* = \arg\max_{v\in\mathcal{V}} \mathcal{I}(v)$.

Hybrid Nearest Frontier and Information Gain (NF + IG). The NF + IG planner scores each candidate viewpoint based on a weighted combination of the navigation distance and the information gain $score(v) = \lambda\mathcal{I}(v) - \mathcal{D}(v)$, where λ is a weighting parameter. It greedily selects the one with the highest score as the next navigation goal: $v^* = \arg\max_{v\in\mathcal{V}} score(v)$. We empirically conduct experiments to tune λ and observe that $\lambda = 20$ performs best on average.

Random Sample (RS). The RS planner randomly selects a viewpoint $v \in \mathcal{V}$ as the next navigation goal.

Learning-based baselines are not considered in this work because their implicit assumptions of the environment setup can hardly be directly applied to our commercialized experimental platforms. In addition, a recent work suggests that above rule-based methods can achieve comparable or even better performance than existing learning-based ones [12].

B. Experiment setup

Simulation environments. We evaluate all methods within Gazebo simulation environments, offering varying degrees of complexity. As depicted in Fig. 3 (a) and (b), we employ two common empty home layouts as foundational testing environments. Fig. 3 (c) - (e) illustrates three fully furnished home layouts, serving as more challenging testing environments. These furnished environments better reflect the practical challenges associated with indoor exploration tasks than previous unfurnished scenarios [10, 32, 12], ensuring a comprehensive assessment of exploration strategies. All simulations are conducted in real-time.

Evaluation metric. We assess exploration efficiency by measuring the total **explored area** within a predetermined time frame, the duration of which is dependent on the environment size. Specifically, we allow $60s$ for Empty home 1, $300s$ for Empty home 2 and Furnished home 1, $250s$ for Furnished home 2, and $200s$ for Furnished home 3.

It is noteworthy that a trained L2E policy can be transferred across robotic platforms. This is because the action space represents high-level navigation goals, rather than specific motor commands, meaning that the low-level planning and execution modules are decoupled from the exploration strategy.

C. Results and analysis

We demonstrate the generalization capability of L2E by training it on the *Furnished home 2* map and testing the resulting policy in all environments. Table I presents the results, with each row representing one home layout. The **area** column signifies the explored area within a fixed exploration time—the higher, the better. Our approach consistently surpasses all baselines, with the **improve** column under each baseline highlighting the degree of improvement afforded by L2E relative to that method. The last column (**minimum improve**) denotes the efficiency enhancement achieved by L2E relative to the most formidable rule-based baseline.

The experiments yield several intriguing observations. Firstly, L2E exhibits more substantial efficiency improvements in

(a) Empty home 1 (b) Empty home 2 (c) Furnished home 1 (d) Furnished home 2 (e) Furnished home 3

Fig. 3: Illustration of the maps.

TABLE I: Comparison between L2E (our method) and other baselines. For each baseline method, the first column (**area**) is the exploration area given a fixed exploration time, which is the higher, the better; the second column (**improve**) is the improvement of our L2E method w.r.t this baseline approach. The last column is the **minimum improvement** against the strongest rule-based baseline.

	Nearest Frontier (NF)		Information Gain (IG)		NF + IG ($\lambda = 20$)		Random Sample (RS)		L2E (ours)	Minimum
	area (m^2)	improve	area (m^2)	improve	area (m^2)	improve	area (m^2)	improve	area (m^2)	improve
Empty home 1	36.33	7.51%	26.87	45.37%	34.55	13.05%	28.29	38.07%	**39.06**	7.51%
Empty home 2	195.27	7.03%	157.00	33.00%	200.95	4.01%	86.07	142.83%	**209.00**	4.01%
Furnished home 1	61.27	40.71%	69.76	23.58%	70.45	22.37%	29.86	188.71%	**86.21**	22.37%
Furnished home 2	73.29	14.97%	70.48	19.55%	71.44	17.95%	35.65	136.35%	**84.26**	14.97%
Furnished home 3	86.46	32.55%	74.09	54.68%	94.77	20.92%	50.95	124.93%	**114.60**	20.92%

complex environments. This is evident from the larger minimum improvements recorded in furnished environments relative to simpler ones, underlining L2E's potential in managing complex real-world scenarios. Secondly, despite being trained in a single map layout, L2E displays commendable generalization across different environments. Finally, no rule-based method consistently outperforms others across all environments. While the NF planner fares best among the baselines in the Empty home 1 and Furnished home 2 environments, the hybrid NF + IG exploration strategy prevails in the remaining environments. This suggests that strategies based on greedy viewpoint selection using meticulously hand-tuned cost functions may not generalize to all scenarios, whereas our learned policy, trained in one environment with a simple and intuitive reward function, consistently outperforms all baselines.

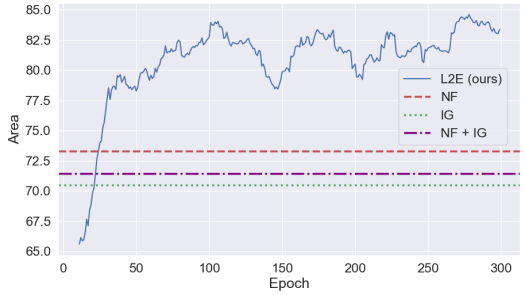


Fig. 4: Training curves in the Furnished home 2 environment.

Fig. 4 shows the training curve of our method and the performance of baseline approaches in the Furnished home 2 environment. From the initial values, we can observe that the performance after IL training is still worse than rule-based ones. However, after a few training epochs, the performance of L2E it quickly surpasses the baselines'. Note that each training epoch corresponds to around 4 minutes of interaction data, which shows the great sample efficiency of the proposed exploration framework and the training algorithm.

Our proposed exploration architecture has full coverage guarantees because of our action space design – it can complete the exploration in a closed indoor environment given enough time. In addition, the entire training procedure is collision-free and safe for the learning robot, because the viewpoint generation phase in Sec. II-A is guaranteed to provide navigable and safe viewpoint candidates. Note that the completeness and safety properties can hardly be achieved by the DRL approaches in the literature.

TABLE II: Ablation study by removing features in L2E.

	Empty home 1	Empty home 2	Furnished home 1	Furnished home 2	Furnished home 3
full algorithm	**39.06**	**209.00**	**82.62**	**84.26**	**114.60**
no pretraining	38.41	201.04	71.03	74.73	107.83
no entropy	35.81	199.81	63.95	74.25	98.75

We also conduct an ablation study of the L2E method by removing the IL pretraining phase or the maximum entropy objective respectively. The result is shown in Table II, where each number corresponds to the explored area. We can see that both the pretraining phase and the entropy term are crucial in learning an efficient exploration policy. Particularly, the entropy term plays a more important factor because it can prevent the policy from being trapped into a local optimum and stopping to explore other more rewarding actions.

IV. CONCLUSION

We propose the learning to explore (L2E) method for indoor autonomous exploration applications. By utilizing a decoupled three-stage exploration framework, we can handle different robot configurations in the viewpoint generation phase, and provide safe and feasible viewpoint candidates for the learning agent as the action space to choose. We show that the proposed method consistently outperforms rule-based exploration strategies in diverse indoor scenarios, showing the advantages of the L2E approach.

References

[1] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4263–4270.

[2] C. Wang, K. V. Hindriks, and R. Babuska, "Active learning of affordances for robot use of household objects," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 566–572.

[3] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.

[4] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. v. Stryk, "Hector open source modules for autonomous mapping and navigation with rescue robots," in *Robot Soccer World Cup*. Springer, 2013, pp. 624–631.

[5] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, "Multi-objective exploration and search for autonomous rescue robots," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 763–777, 2007.

[6] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-d environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[7] C. Leung, S. Huang, and G. Dissanayake, "Active slam using model predictive control and attractor based exploration," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5026–5031.

[8] ——, "Active slam in structured environments," in *2008 IEEE International conference on Robotics and Automation*. IEEE, 2008, pp. 1898–1903.

[9] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[10] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng, "Deep reinforcement learning supervised autonomous exploration in office environments," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7548–7555.

[11] "Amazon astro." [Online]. Available: https://www.amazon.science/tag/amazon-astro

[12] Y. Xu, J. Yu, J. Tang, J. Qiu, J. Wang, Y. Shen, Y. Wang, and H. Yang, "Explore-bench: Data sets, metrics and evaluations for frontier-based and deep-reinforcement-learning-based autonomous exploration," *arXiv preprint arXiv:2202.11931*, 2022.

[13] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[14] Y. Niu, R. Paleja, and M. Gombolay, "Multi-agent graph-attention communication and teaming," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 2021, pp. 964–973.

[15] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[16] C. Yu, X. Yang, J. Gao, H. Yang, Y. Wang, and Y. Wu, "Learning efficient multi-agent cooperative visual exploration," *arXiv preprint arXiv:2110.05734*, 2021.

[17] A. R. Dooraki and D. J. Lee, "Memory-based reinforcement learning algorithm for autonomous exploration in unknown environment," *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, p. 1729881418775849, 2018.

[18] A. Ramezani Dooraki and D.-J. Lee, "An end-to-end deep reinforcement learning-based intelligent agent capable of autonomous exploration in unknown environments," *Sensors*, vol. 18, no. 10, p. 3575, 2018.

[19] M. Keidar and G. A. Kaminka, "Robot exploration with fast frontier detection: Theory and experiments," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 113–120.

[20] K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager, "Uncertainty-aware occupancy map prediction using generative networks for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5453–5459.

[21] Z. Liu, B. Chen, H. Zhou, G. Koushik, M. Hebert, and D. Zhao, "Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 748–11 754.

[22] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "Tnt: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.

[23] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2095–2104.

[24] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.

[25] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition, 2021, pp. 14 403–14 412.

[26] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.

[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[28] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[30] B. Eysenbach and S. Levine, "Maximum entropy rl (provably) solves some robust rl problems," *arXiv preprint arXiv:2103.06257*, 2021.

[31] F. Chen, P. Szenher, Y. Huang, J. Wang, T. Shan, S. Bai, and B. Englot, "Zero-shot reinforcement learning on graphs for autonomous exploration under uncertainty," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5193–5199.

[32] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.