Motion Planning using Transformers

Jacob J. Johnson[1] and Michael Yip[1]

***Abstract*— Motion planning is integral to robotics applications such as autonomous driving, surgical robots, and industrial manipulators. Existing planning methods lack scalability to higher-dimensional spaces, while recent learning-based planners have shown promise in accelerating sampling-based motion planners (SMP) but lack generalizability to out-of-distribution environments. To address this, my research explored the use of Transformers for motion planning. We proposed two planning techniques - Motion Planning Transformers (MPT) and Vector Quantized-Motion Planning Transformers (VQ-MPT)- that overcome previous learning-based methods' generalization and scaling drawbacks. Both methods split large planning spaces into discrete sets and selectively choose the sampling regions. This enables the planners to accelerate and integrate with out-of-the-box SMPs while generating near-optimal paths. Trained models of MPT and VQ-MPT generalize to environments unseen during training and achieve higher success rates than previous methods. MPT can generalize to costmaps of varying sizes while VQ-MPT is generalizable in that it can be applied to systems of varying complexities, from 2D mobile to 14D bi-manual robots with diverse environment representations, including costmaps and point clouds.**

I. INTRODUCTION

Sampling-based motion planning uses randomly sampled points to generate a tree-based collision-free path between a start and goal location [1], [2]. However, random sampling is inefficient [3] for goal-directed tasks, particularly when the search space spans a high number of dimensions. Since sampling-based motion planners (SMPs) are a fundamental component of numerous autonomous systems [4], [5], improving the efficiency and generalizability of the underlying planners enables these systems to handle more complex tasks with intricate sequences of planning, improves task execution, and reduces the need to re-parametrize planners for different environments.

While SMPs effectively find a trajectory, they face several challenges in improving sampling efficiency. As the dimensionality of the planning space increases, the "curse of dimensionality" makes sampling more difficult and time-consuming. Efficiently exploring planning spaces to find feasible paths is also a significant challenge. The parameters of these planners also need to be reconfigured to solve for different environments reliably. Most of these planners are probabilistically complete, i.e., the planner will find a path if a trajectory exists, given enough time. But finding an optimal trajectory, like the shortest path, is challenging, especially for higher dimensional spaces. Numerous works have been proposed that address some combinations of these challenges.

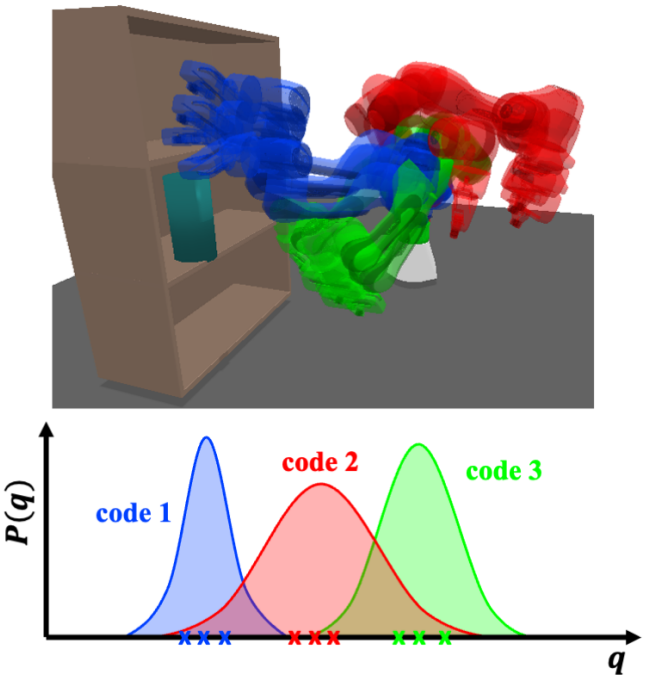

Fig. 1. VQ-MPT can efficiently split high-dimensional planning spaces into discrete sets of distributions. Each distribution is represented using a latent variable called code or dictionary value. Given a planning problem, the model selects a subset of codes and samples from the associated distributions to construct the trajectory. By sampling efficiently, VQ-MPT reduces planning times by 2-6× compared to previous planners.

Recently, the integration of point cloud data has gained traction in planning methodologies, enabling the capture of diverse scenes with intricate environment representations [6], [7]. Planning using sensory data such as costmaps and point clouds can have several benefits. Point clouds allow for a more accurate environment representation, enabling precise motion planning as the robot can better understand the surroundings, including complex shapes and uneven surfaces. For robotic arms and manipulators, point clouds can aid in a better understanding of object shapes and sizes, leading to improved grasping and manipulation strategies. This is especially useful in tasks like pick-and-place operations. Models that rely on point clouds have also been shown to be easier for real-to-sim transfer [8] because they rely on the geometry of the points rather than the texture and pixels from an image. Therefore, planning models that utilize point cloud data are valuable to the robotics community.

Numerous learning-based methods have been proposed before that address the challenges of SMP, utilizing envi-

[1]J.J. Johnson and M.C.Yip are with the Electrical and Computer Engineering Department at the University of California San Diego, La Jolla, CA, USA {jjj025, yip}@eng.ucsd.edu

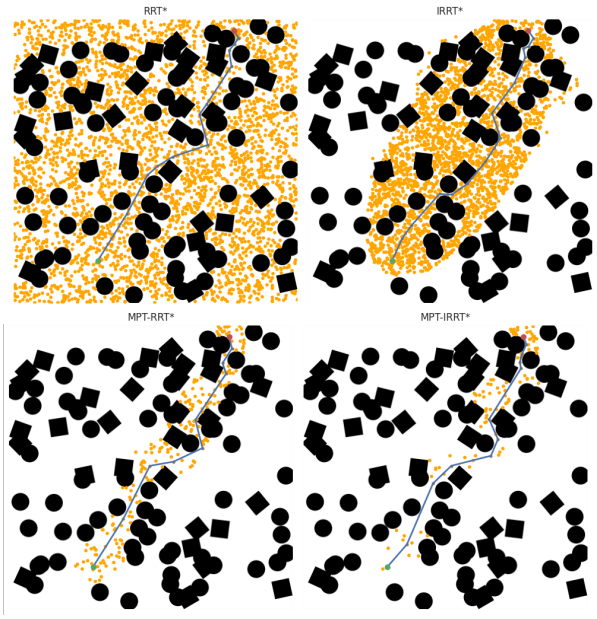

Fig. 2. (From top left clockwise) Vertices used by RRT*, IRRT*, MPT-IRRT*, and MPT-RRT* for the same start (green) and goal (red) positions. MPT aided planners can significantly reduce the number of vertices (orange) required to search for a path.

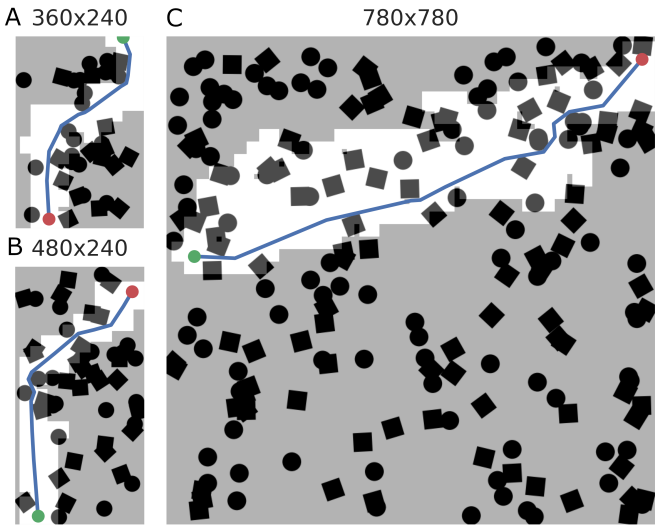

Fig. 3. Plots of MPT aided planning for out-of-distribution environments. A, B, C: Plot of paths for Random Forest environments of different sizes. The architecture of the MPT Model allows flexibility in planning for environments of different sizes.

ronment representations such as point clouds and costmaps [9], [10], [11], [12], [13], [14]. These methods learn from previously planned paths and use this experience to efficiently plan for new scenes and environments. Some of these methods also scale well to high-dimensional planning spaces. However, poor generalization makes these models intractable for real robotics applications.

Recent advances in large language models, such as BERT [15], and GPT [16], have inspired similar efforts in solving planning tasks using transformer models [17], [18], [19]. Transformer models are an ideal candidate for solving planning tasks because of their ability to make long-horizon connections [20]. These models make better control decisions in robotic quadrupedal walking tasks by attending to proprioceptive and visual sensor data [21]. [19] propose transformer models for solving for planar manipulators and 2D mobile robots. Although these works support the possibility of using transformer models for motion planning, for models that attend to sensory data, it is difficult to interpret the policy's future control actions and provide any form of guarantee for the underlying planner.

Our initial work, Motion Planning Transformers [22], demonstrated that transformer models could be applied to the motion planning problem and significantly accelerate sample-based planning; however, via direct application of transformer models developed for 2D data such as images and videos, the scope of planning was limited to 2D maps that would be only conducive to solving planar robots and vehicle planning problems. Our follow-up paper introduced Vector Quantized-Motion Planning Transformer (VQ-MPT) [23], a transformer-based model that uses vector quantization to discretize the planning space into a set of distributions. Through our experiments, we show, by improving sampling efficiency, how transformer-based models reduce planning time compared to previous heuristics-based methods such as BIT*[24] and improve generalization to unseen out-of-distribution environments than learned planners such as Motion Planning Networks (MPNet) [9].

II. METHODS & RESULTS

A. Motion Planning Transformers

Transformer models have been extensively used for image reconstruction tasks [25], [26] due to their ability to make long-horizon correlations. Our first work, Motion Planning Transformer (MPT), utilized Vision Transformers [27] for planning for 2D mobile systems where environments are represented as costmaps [22]. Each costmap was discretized into grids, and a region proposal network using a transformer architecture attended to different patches to identify regions of interest for the current planning problem. Once a particular region has been identified, an off-the-shelf SMP was used to identify regions of interest for the current planning problem. We also proposed a novel positional encoding while training that enables the trained model to generalize to maps of different sizes (Fig. 3). We also expanded our framework for planning for $SE(2)$ robots, where the region proposal network also predicts an orientation for each grid it selects, and the SMP planner samples a random pose from a Gaussian distribution centered at the predicted pose.

By reducing the search regions, we show that MPT-aided planners reduce vertices on the planning tree by 2-12 $\times$ and planning time by 7-25$\times$ compared to traditional planners for 2D robots (See Fig. 2 while for $SE(2)$ robots, the planner can improve planning time by 2$\times$. Due to

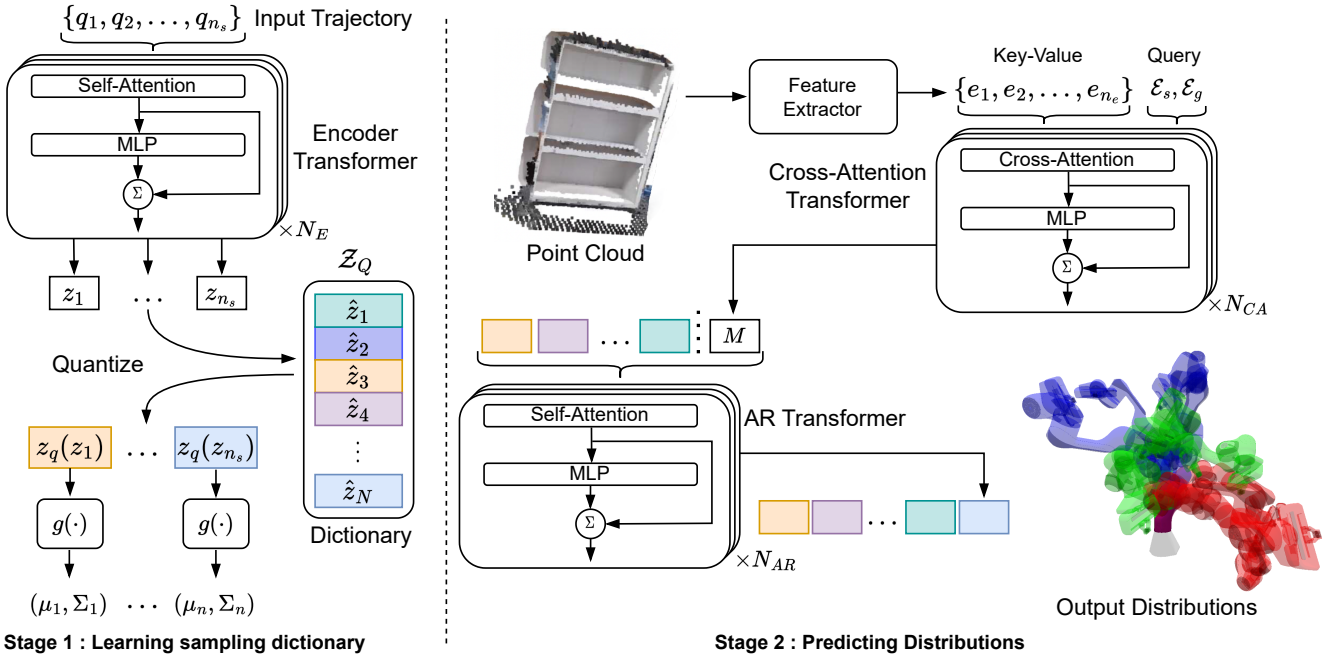

Fig. 4. An outline of the model architecture of VQ-MPT. Stage 1 (Left) is a Vector Quantizer that learns a set of latent dictionary values that can be mapped to a distribution in the planning space. By encoding the planning space to discrete distributions, we can plan for high-dimensional robot systems. Stage 2 (Right) is the Auto-Regressive (AR) model that sequentially predicts the sampling regions for a given environment and a start and goal configuration. The cross-attention model transduces the start and goal embeddings given the environment embedding generated using a feature extractor. The output from the AR Transformer is mapped to a distribution in the planning space using the decoder model from Stage 1.

their ability to make long-horizon correlations, MPT-assisted planners achieve a 7-28% improvement in accuracy over recent learning-based planners. Our novel position encoding improves the planner's accuracy by 60% for larger maps. We also provide a ROS2 plugin for the Nav2 navigation stack [28] for our method. This will benefit the robotics community to work with and extend our models for planning.

Although MPT improved planning for mobile robots, two challenges prevented it from extending to higher-dimension planning spaces. First, MPT required splitting the entire planning space into discrete grids. Hence, for the 7D robot, if each dimension is split into 10 segments, that's a million grids for the region proposal network to attend. Hence, applying transformers to this type of discretization would be computationally intractable. Secondly, MPT requires the planning and task space to overlap, but this is true for manipulation systems where the task space is $SE(3)$ while the planning space is $\mathbb{R}^n$. Our follow-up work on Vector Quantized-Motion Planning Transformers (VQ-MPT) tackles these challenges.

B. Vector Quantized-Motion Planning Transformer

An outline of VQ-MPT is given in Fig. 4. The model consists of two stages - a quantization stage and a prediction stage. The quantization stage segments the planning space as a collection of distributions rather than discretized grids. VQ-MPT uses a Vector Quantized (VQ) model to generate the collection of distributions of the planning space. VQ models are generative models with an encoder-decoder architecture similar to Variational AutoEncoder (VAE) models but with the latent dimension represented as a collection of learnable vectors referred to as dictionaries. Each dictionary value represents a distribution in the planning space.

The prediction stage generates sampling regions by predicting indexes from the dictionary set for a given planning problem and sensor data. It comprises two models - a cross-attention model to embed start and goal pairs and the environment embedding into latent vectors (M) and a transformer-based Auto-Regressive (AR) model to predict the dictionary indexes. The environment representation (i.e., costmap or point cloud data) is passed through a feature extractor to construct the environment encodings $\mathcal{E} = \{e_1, e_2, \ldots, e_{n_e}\}$ where $e_i \in \mathbb{R}^d$. The feature extractor reduces the dimensionality of the environment representation and captures local environment structures as latent variables using convolutional layers for costmaps and set-abstraction proposed in PointNet++ [29] for point clouds. We chose these architectures because they are agnostic to the environment size and can generate latent embeddings for larger-sized costmaps or point clouds. The start and goal states (q_s and q_g) are projected to the start and goal embedding ($\mathcal{E}_s \in \mathbb{R}^d$ and $\mathcal{E}_g \in \mathbb{R}^d$) using a MLP network. The cross-attention model uses the environment embedding, $\mathcal{E}$, and the start and goal embedding, $\{\mathcal{E}_s, \mathcal{E}_g\}$ to generate latent vectors M. The cross-attention model learns a feature embedding that fuses the given start and goal pair with the given planning environment. It uses the vector in $\mathcal{E}$ as key-value pairs, and $\mathcal{E}_s$ and $\mathcal{E}_g$ as query vectors to generate M. Thus, learning a latent representation that combines the task and planning space.

TABLE I

PLANNING STATISTICS FOR OUT-OF-DISTRIBUTION ENVIRONMENTS

Robot		IRRT*	BIT*	RRT	MPNet	VQ-MPT
7D	Accuracy	44.60%	37.80%	84.20%	53.20%	**92.20%**
	Time (sec)	55.12	75.32	8.88	10.14	**3.24**
	Vertices	215	5147	477	310	306
14D	Accuracy	10.60%	12.20%	75.00%	80.40%	**98.60%**
	Time (sec)	20.72	30.07	19.75	23.91	**6.21**
	Vertices	20	1673	179	104	**70**
7D (Real)	Accuracy	10/10	10/10	10/10	3/10	10/10
	Time (sec)	30.68	26.42	1.69	2.23	**1.17**
	Vertices	607	2852	**21**	7	34

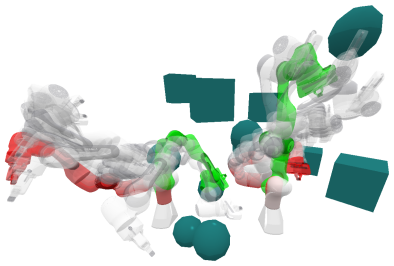

Fig. 5. Sample paths planned by the VQ-MPT planner for 14D robot on an in-distribution environment. The red and green color represents the start and goal states of the robot, respectively. Given an environment with crowded obstacles, VQ-MPT can sample efficiently from learned distributions to find a trajectory.

Fig. 6. Snapshots of a trajectory planned using VQ-MPT on a physical Panda robot. VQ-MPT generalizes to real-world sensor data without any additional data collection or fine-tuning and reduces planning time for finding near-optimal paths.

We evaluated our framework on a simulated 7D Franka Panda Arm and a 14D Bimanual robot setup (See Fig. 5). VQ-MPT models for both these robots were trained using RRT* trajectories and point cloud data generated in simulation. Our experiments compared the use of VQ-MPT coupled with RRT planner with traditional and learning-based planners on a diverse set of planning problems. All planners were implemented using the Open Motion Planning Library (OMPL) [30].

The environments the models were trained on are similar to those seen in Fig. 5, where obstacles are placed randomly. The environments used for testing consisted of real-world obstacles such as shelves and cupboards as in Fig. 1. Our results show that VQ-MPT can efficiently discretize the planning space and select sampling regions to construct optimal trajectories. By leveraging sensor data such as point clouds, VQ-MPT can narrow down the search region in the planning space, enabling it to achieve 8-24% more accuracy than non-optimal planners such as RRT.

To evaluate the performance of VQ-MPT on physical sensor data, we tested a trained model in a real-world environment (Fig. 6). The environment was represented using point cloud data from Azure Kinect sensors, and collision checking was done using the octomap collision checker from Moveit [1]. Camera to robot base transform was estimated using markerless pose estimation technique [31]. Our results in Table I show that the model can plan trajectories faster than RRT with the same accuracy. This experiment shows that VQ-MPT models can also generalize well to physical sensor data without further training or fine-tuning. Such generalization will benefit the larger robotics community since other researchers can use trained models in diverse settings without collecting new data or fine-tuning the model.

[1]https://moveit.ros.org/

III. FUTURE WORK & CONCLUSION

Our work has explored the benefits of using transformers for reducing the search spaces for SMP's. Both MPT and VQ-MPT generalize to a wide array of environments outside their training set, making it easier to disseminate trained models to the wider robotics community. VQ-MPT, in theory, could scale to larger dimensions, making the approach applicable to a wide range of robot systems. Using feature extractors such as PointNet++, our model better understands geometric objects, enabling robust transfer of these models to real-world systems.

Our work has laid the groundwork for using transformers for planning. The dictionary encodings in VQ-MPT could be considered the fundamental building blocks for setting up a robot language. In our work, we solved the motion planning problem by selecting a set of these words to construct the robot trajectory. More complex problems, such as task and motion planning (TAMP), require stitching together several trajectories, which could be thought of as composing a paragraph. Given the recent ability of large language models such as GPT [16] to generate complex phrases, a promising research direction would be to explore the use of VQ-MPT for solving TAMP problems. Other works could explore extending VQ-MPT for constraint and kinodynamic planning.

REFERENCES

[1] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, 2001.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Auto.*, 1996.

[3] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *IEEE Int. Conf. on Robotics and Auto.*, 2003.

[4] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning," in *2021 IEEE Int. Conf. on Robotics and Auto. (ICRA)*, 2021, pp. 7737–7743.

[5] R. Alterovitz, K. Goldberg, and A. Okamura, "Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles," in *Proceedings of the IEEE Int. Conf. on Robotics and Auto.*, 2005.

[6] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5556–5565.

[7] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016.

[8] X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han, Z. Huang, T. Mu, J. Xu, and H. Su, "Close the optical sensing domain gap by physics-grounded active stereo sensor simulation," 2023.

[9] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, 2020.

[10] J. J. Johnson, L. Li, F. Liu, A. H. Qureshi, and M. C. Yip, "Dynamically constrained motion planning networks for non-holonomic robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[11] L. Li, Y. Miao, A. H. Qureshi, and M. C. Yip, "Mpc-mpnet: Model-predictive motion planning networks for fast, near-optimal planning under kinodynamic constraints," *IEEE Robotics and Auto. Letters*, vol. 6, no. 3, pp. 4496–4503, 2021.

[12] P. Lehner and A. Albu-Schäffer, "The repetition roadmap for repetitive constrained motion planning," *IEEE Robot. and Autom. Letters*, 2018.

[13] C. Chamzas, Z. Kingston, C. Quintero-Peña, A. Shrivastava, and L. E. Kavraki, "Learning sampling distributions using local 3d workspace decompositions for motion planning in high dimensions," in *IEEE Int. Conf. on Robot. and Autom.*, 2021.

[14] R. Kumar, A. Mandalika, S. Choudhury, and S. Srinivasa, "Lego: Leveraging experience in roadmap generation for sampling-based planning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019.

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020.

[17] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems*, 2021.

[18] M. Janner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," in *Advances in Neural Information Processing Systems*, 2021.

[19] D. S. Chaplot, D. Pathak, and J. Malik, "Differentiable spatial planning using transformers," in *ICML*, 2021.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[21] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," in *Int. Conf. on Learning Representations*, 2022.

[22] J. J. Johnson, U. S. Kalra, A. Bhatia, L. Li, A. H. Qureshi, and M. C. Yip, "Motion planning transformers: A motion planning framework for mobile robots," 2021.

[23] J. J. Johnson, A. H. Qureshi, and M. Yip, "Learning sampling dictionaries for efficient and generalizable robot motion planning with transformers," 2023.

[24] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE Int. Conf. Robot. Autom.*, 2015.

[25] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, "Vector-quantized image modeling with improved VQGAN," in *Int. Conf. on Learning Representations*, 2022.

[26] A. van den Oord, O. Vinyals, and k. kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. on Learning Representations*, 2021.

[28] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The marathon 2: A navigation system," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020.

[29] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017.

[30] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Auto. Magazine*, 2012.

[31] J. Lu, F. Richter, and M. C. Yip, "Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer," 2023.