IMPACT: Intelligent Motion Planning with Acceptable Contact Trajectories via Vision-Language Models

Yiyang Ling*, Karan Owalekar*, Oluwatobiloba Adesanya, Erdem Bıyık, Daniel Seita

***Abstract*— Motion planning involves determining a sequence of robot configurations to reach a desired pose, subject to movement and safety constraints. Traditional motion planning finds collision-free paths, but this is overly restrictive in clutter, where it may not be possible for a robot to accomplish a task without contact. In addition, contacts range from relatively benign (e.g., brushing a soft pillow) to more dangerous (e.g., toppling a glass vase), making it difficult to characterize which may be acceptable. In this paper, we propose IMPACT, a novel motion planning framework that uses Vision-Language Models (VLMs) to infer environment semantics, identifying which parts of the environment can best tolerate contact based on object properties and locations. Our approach generates an anisotropic cost map that encodes directional push safety. We pair this map with a contact-aware A* planner to find stable contact-rich paths. Our results over 3200 simulation and 200 real-world trials suggest that IMPACT enables efficient contact-rich motion planning in cluttered settings while outperforming alternative methods and ablations. Our project website is available at https://impact-planning.github.io/.**

I. Introduction

Classical motion planning for robot manipulation [1], [2] frames the problem as finding a path for the robot's end-effector to reach a target while avoiding collisions with obstacles. This formulation is generally desirable, but can be highly restrictive, especially in densely cluttered environments [3]. In such cases, some incidental contact may be necessary to achieve a task, or to accomplish it more efficiently than by strictly avoiding all collisions.

Consider the example in Fig. 1, which shows a robot manipulator making contact with a toy bear to efficiently reach the target spice jar. Due to the clutter, a collision-free path either does not exist or would require a longer parabolic motion to go above the obstacles (which, again, may not be feasible in cluttered cabinets and boxes). We thus desire robots that can achieve a task while moving through a cluttered environment, making appropriate contact as needed.

To address this challenge, we propose IMPACT, a novel framework that leverages modern VLMs such as GPT-4o [4] to infer initial semantic costs for objects in the scene. Unlike prior methods, our approach repeatedly samples push outcomes to estimate the probability of collisions, condensing this risk into an anisotropic cost map where interaction costs vary by direction. We integrate this map with a contact-aware A* planner [5] to find stable, contact-rich paths.

*Equal contribution.

All authors are with Thomas Lord Department of Computer Science, Viterbi School of Engineering, University of Southern California, USA.

Correspondence: {lingyiya,kowaleka,seita}@usc.edu.

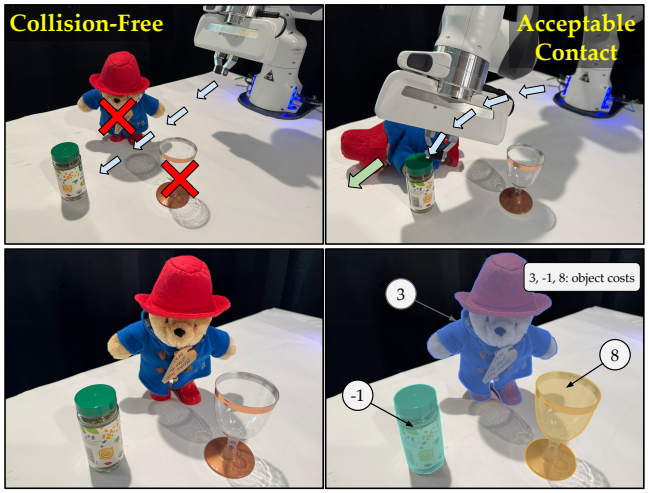

Fig. 1: An example of a reaching task and object costs. The first row shows the difference between collision-free paths and paths with acceptable contact. Left: Collision-free paths prevent a "straight" path to the spice jar because of the toy bear and wine glass obstacles (each marked with a red "X"). Right: With semantically acceptable contact, the robot can successfully reach the spice jar by pushing the toy bear and avoiding the fragile wine glass. The second row shows the cost of each object generated by GPT-4o. Left: the original scene. Right: GPT-4o assigns different costs to objects, with the target assigned -1 (toy bear: 3, spice jar: -1, wine glass: 8).

II. Related Work

Classical motion planners [6]–[9] prioritize avoiding all collisions, which is often infeasible in dense clutter. While prior works such as Navigation Among Movable Obstacles [10] reason about relocating obstacles, others treat certain obstacles as permeable to allow for incidental contact [11], [12].

Our work is most closely related to language-conditioned planning; however, methods like LAPP [13] and VoxPoser [14] require explicit user instructions regarding which objects are safe to touch (e.g., *"can collide with toys"*). In contrast, IMPACT leverages the inherent commonsense knowledge within VLMs to automatically infer these tolerances without explicit guidance, enabling more autonomous manipulation in cluttered environments.

Our contributions are as follows:

- IMPACT, a framework that formalizes "acceptable contact" by transforming initial VLM-inferred semantic costs into a dense, anisotropic cost map to represent the directional safety of physical interactions.
- A contact-aware A* planner capable of interpreting this anisotropic map to execute paths with intelligent and

minimal-impact contact.
- Extensive simulation and real-world experiments, including a human subjects study, that validate our approach using both objective and subjective metrics to evaluate the success of contact-rich trajectories.

III. Problem Statement

We assume a single robot arm with a standard gripper operates in a densely cluttered environment that contains n objects, denoted as $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$. The robot must reach a given target object $o_{\text{targ}} \in \mathcal{O}$ while minimizing unwanted contact with other objects (i.e., obstacles) in $\mathcal{O} \setminus o_{\text{targ}}$. In environments with significant clutter, o_{targ} may be behind or close to multiple objects and thus reaching it may be infeasible with a collision-free trajectory. Given RGBD image observations from at least two cameras, the objective is to compute a trajectory τ for the robot, defined as a sequence of end-effector poses, such that its end-effector ultimately touches o_{targ} while engaging in "semantically-acceptable" contact with obstacles when needed.

IV. Method: IMPACT

Our framework consists of two main steps (see Fig. 2). First, it uses a VLM to obtain object costs in a cluttered scene (Sec. IV-A), and then uses those costs for contact-rich motion planning (Sec. IV-B).

A. Obtaining Object Costs using a VLM

A key technical challenge is defining the notion of an "acceptable" contact. This depends heavily on semantics, or the general-purpose commonsense knowledge that humans have about the behavior of diverse objects. Different objects with varying materials, geometries, sizes or purposes have different tolerances to potential contact. Furthermore, tolerance to contact should also depend on an object's proximity to nearby objects. Therefore, we leverage the commonsense knowledge in VLMs to estimate the tolerance rate for contact of each object. We encode this information by assigning each object to an integer in $\{0, 1, \ldots, 10\}$ as the cost, where a higher cost indicates lower tolerance to contact with any part of the robot arm as it executes a trajectory. For example, the cost of a fragile object (e.g., wine glass) should be significantly higher compared to the cost of an object that can better absorb contact (e.g., foam rubber).

We use GPT-4o (hereafter, GPT) as the VLM to generate the cost of all objects in the scene due to its strong spatial reasoning capabilities [16], but our approach is compatible with other VLMs. The VLM is used in a zero-shot inference setting without any fine-tuning. The input to the VLM includes both an annotated image and a text prompt. We mount an RGBD camera to capture a scene image I, and use SAM2 [15] to segment and annotate the objects. The segmented and annotated image I' is ultimately provided as part of the input to GPT. We also design a text prompt template ℓ which includes the list of objects in the scene (but is otherwise task-agnostic), and some general principles related to the concept of contact tolerance. The output of GPT is a dictionary with the cost of all queried objects. Fig. 1 (second row) shows an example of object costs generated from GPT. It assigns a high cost of 8 to the wine glass and a lower cost of 3 to the toy bear, which indicates different tolerances to contact across objects.

B. Directional Contact-Aware Motion Planning

We use motion planning algorithms to synthesize robot trajectories. We construct a 3D voxel grid V where each voxel $V[x, y, z]$ denotes the cost at position (x, y, z). During planning, we assign all voxels corresponding to the target object a cost of -1 to encourage the planner to find a path to it. To evaluate directional contact information, we project V into a 2D grid M using a top-down view, where $M[x, y] = \max_z V[x, y, z]$. From this 2D perspective, all obstacles are divided into low-cost (5 or lower) and high-cost (6 or higher) sets based on their VLM-assigned costs. To simplify planning, we define three motion primitives in 2D space: (i) *Move*, which translates the robot to one of eight neighboring positions; (ii) *Rotate*, which discretizes the end-effector orientation changes within $\pm 45°$; and (iii) *Push*, which moves the end-effector within a bounded radius while making contact with an object. We assume the first two change the end-effector's pose without affecting the environment. In contrast, the Push primitive denotes end-effector translations that involve some robot-object contact.

1) Anisotropic Cost Map Generation: The map M encodes object costs independent of motion direction. In clutter, however, contact safety depends strongly on the direction where the robot approaches an object. To address this, we construct a new **anisotropic cost map** M' that augments each point with directional safety information. Defining M' over both 2D positions and 2D directions would be high-dimensional and computationally expensive, so we keep M' as a 2D function. For each point (x, y) on the boundary of a low-cost object, we consider the *reverse surface normal* vector, which points "inwards" towards the object. This gives the most natural and direct push for the object, but to consider different possible pushes, we sample m push outcomes with small variations in distance and angle, weighted by a 2D Gaussian centered on that normal (so outcomes closer to that vector receive higher likelihood). Each hypothesized outcome is evaluated on M by checking whether the displaced object would overlap with other objects, and classified into four categories: safe, low-cost contact, high-cost contact or contact with the target. Push outcomes in the same category are assigned an identical safety score $u \in (0, 1]$, with larger values indicating safer outcomes. The aggregated safety score at (x, y) is:

$$f_s(x, y) = \frac{\sum_{i=1}^{m} l_i u_i}{\sum_{i=1}^{m} l_i}, \tag{1}$$

where u_i encodes the safety level, and l_i is the likelihood of the i-th push outcome. This score is then used to update the cost map at (x, y), creating the final anisotropic cost map M' where each point has an associated directional push score:

$$M'[x, y] = \alpha M[x, y] + (1 - \alpha)[10 - 10 f_s(x, y)]. \tag{2}$$

Category	Path Planning Algorithm	Reach Target ↑	Path Cost ↓	Contact Duration (s) ↓	Unsafe Object Displacement (cm) ↓	Success Rate ↑
Collision Free	RRT	66.75%	-	**1.12**	6.47	28.75%
	RRT*	61.00%	-	1.25	7.78	26.50%
	A* w/o direction	20.00%	-	8.33	6.51	15.00%
	A*	23.33%	-	5.14	**1.93**	15.75%
VLM Cost	RRT	62.91%	9.04	1.22	7.84	57.25%
	RRT*	59.50%	9.11	1.26	7.80	50.75%
	A* w/o direction	58.25%	6.71	5.93	3.82	57.50%
	A* (IMPACT)	**78.00%**	**5.93**	4.18	2.51	**73.75%**
-	LAPP	50.00%	-	9.37	9.22	25.00%

TABLE I: Comparison of path planning algorithms and results in PyBullet simulation. "Success Rate" only counts the trajectories that reach the target and satisfy thresholds on "Path Cost," "Contact Duration," and "Unsafe Object (human-designated) Displacement." The arrow ↑ indicates larger values of the metric correspond to better performance, and ↓ represents the opposite. Collision-free baselines and LAPP [13] do not use VLMs to generate object costs, so they do not have values for "Path Cost" in this table.

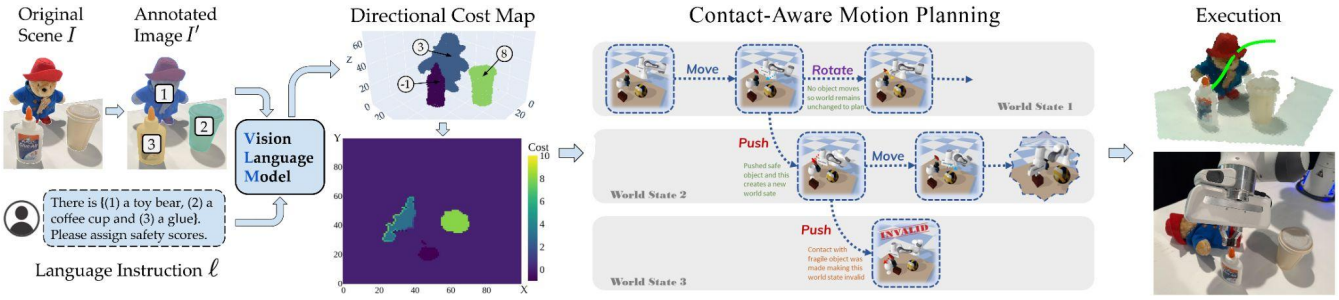

Fig. 2: Overview of IMPACT. There is a toy bear, a coffee cup and a glue bottle (target) on the table. The VLM receives an annotated image I' and a language template prompt ℓ with object information from SAM2 [15], and outputs *costs* for the three objects. We use a cost of -1 for the target object. We construct a 3D voxel grid V using these costs and then flatten it to produce an anisotropic, contact-aware cost map M'. The contact-aware A* planner searches over three motion primitives in this map: *Move*, *Push* and *Rotate* to generate a trajectory. The planner's state space includes the robot's end-effector pose and the displaced positions of low-cost objects. These guide the robot to avoid the coffee cup but make contact with the toy bear at the appropriate direction to reach the glue bottle.

Here, $\alpha \in (0, 1)$ is a weight parameter to balance the object cost and the directional safety score $f_s \in [0, 1]$, and the constant 10 ensures the updated cost in M' remains consistent with the original object cost range in M.

2) Contact-Aware A Planning:* To generate a plan, we search for a sequence of primitives that guides the end-effector to the target while accounting for directional costs in M'. We formulate this as a graph search problem and use A* [5], which is well-suited for this setting and provides deterministic control compared to sampling-based algorithms. The planner's state is defined as $S = (p, r, \mathcal{D})$, where (p, r) is the robot end-effector position and orientation in 2D space and $\mathcal{D}$ is a list that tracks the cumulative 2D displacement of low-cost objects. At each step, the planner expands nodes using the three primitives. For Push, we estimate the object displacement using $d = D \cdot \big(\cos(\varphi)\big)^{\gamma}$, where D is the maximum displacement when pushing in the optimal direction, φ is the angular deviation between the push direction and the object's reverse surface normal at the contact point, and the decay parameter γ is set to 1.3. IMPACT extends standard A* search by incorporating world state changes caused by pushes. Each valid Push creates a new branch in the search tree where an object has been permanently displaced to a new position. The resulting state, $S' = (p', r', \mathcal{D}')$, contains the updated robot pose and an updated world configuration.

A* search evaluates each state s using the cost function $f(s) = g(s) + h(s)$, where $g(s)$ is the accumulated path cost from the start to state s, and $h(s)$ is a heuristic estimate of the remaining cost to the goal. We use the standard Euclidean distance to the target object as the heuristic $h(s)$. The accumulated path cost $g(s')$ is updated at each step with action $a : s = (p, r, \mathcal{D}) \to s' = (p', r', \mathcal{D}')$ as:

$$g(s') = g(s) + C(a) + P(s'), \tag{3}$$

where $C(a)$ is the action cost and $P(s')$ is a gripper placement penalty. We define action costs based on the primitive. Move assigns a cost proportional to the Euclidean distance between two end-effector positions. Rotate applies a fixed cost for orientation changes. Push generates its cost from the value $M'[x, y]$ at the contact point (x, y). The penalty $P(s')$ consists of three terms: (i) the average cost of the region the gripper overlaps, (ii) a collision penalty if the gripper collides with an object, and (iii) a proximity penalty proportional to the distance to high-cost objects. By combining these costs, the planner can find a path that minimizes risk while permitting acceptable contact when necessary. The path is then converted back to 3D space for evaluation and execution.

V. Experiments

A. Simulation Setup and Baselines

We build and test our pipeline using PyBullet simulation [17]. We create 20 scenes designed to test contact-rich manipulation, using a hybrid object dataset which includes tall and bulky items (e.g., sugar boxes and water pitchers) from the YCB [18] dataset and fragile objects (e.g., wine glasses and stacked bowls) using 3D models generated from TRELLIS [19]. All objects are placed on a tabletop. We use three cameras to capture the scene for cost map generation: one in front of the scene and two on the sides.

We compare IMPACT against several baselines: (i) *Collision-Free Planning* using RRT, RRT*, and A* variants, which avoids all collisions and serves as a baseline to demonstrate that in cluttered environments, a collision-free path may not exist; (ii) VLM Cost with RRT [8] and RRT* [9] as planners on the 3D cost map V; (iii) VLM Cost+A* w/o direction, which uses isotropic costs without directional push analysis; and (iv) LAPP [13], which trains a language-conditioned collision function using CLIP [20] embeddings to allow collisions with specific objects specified via language instructions.

We define a trajectory as a *success* if the robot reaches the target with path cost < 10, contact duration < 100s, and displacement < 10cm for all human-designated unsafe obstacles. A human annotator pre-selects the 1-2 highest-cost objects in each scene as unsafe for all evaluated methods.

B. Simulation Results

Table I reports our simulation results. IMPACT achieves the highest success rate with lower path cost, shorter contact duration, and smaller displacement of unsafe objects (pre-selected by human). Paths planned by Collision-Free baselines can also lead to unexpected collisions. These baselines prioritize avoiding all obstacles, but the robot may lack continuous feasible joint configurations to execute the path and becomes too close to obstacles, increasing the risk of collisions. IMPACT also outperforms the VLM Cost+A* w/o direction, which we attribute to the contact-aware information embedded in the anisotropic cost map. The LAPP baseline has a lower `reach_target` and success rate, mainly because it sometimes fails to find paths or push obstacles along the path. Table II shows that VLM-assigned costs consistently improve success rate across all planners compared to setting all costs to 0.

TABLE II: Ablation study on object costs. "Same cost for all" assigns all object costs to 0 instead of querying the VLM.

Planner	Cost	Success Rate
RRT	VLM Cost	**57.25%**
RRT	Same Cost for All	44.50%
RRT*	VLM Cost	**50.75%**
RRT*	Same Cost for All	42.50%
A*	VLM Cost (IMPACT)	**73.75%**
A*	Same Cost for All	65.00%

We also conduct a human evaluation with 25 participants, each answering 20 questions comparing trajectories from one method against some baseline. Across all comparisons, the method using VLM-assigned costs is the most preferred, suggesting that cost maps produce motion plans that better align with human preferences by leveraging commonsense knowledge in the VLM.

TABLE III: Results of our method versus LAPP in the real world. Seen objects refer to objects seen during fine-tuning of LAPP while unseen objects are novel objects to LAPP. Our method is zero-shot, so all the objects can be considered unseen to it.

Method	Success Rate	
IMPACT	**61%**	
LAPP	49% (Seen Obj.)	40% (Unseen Obj.)

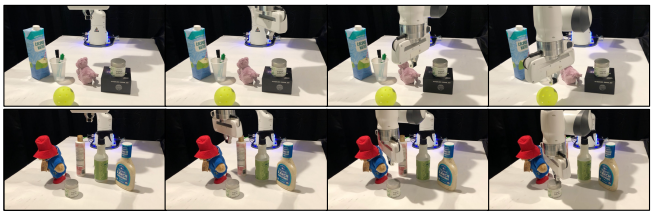

Fig. 3: Examples of successful trajectories from IMPACT in the real world. Top row: the robot reaches the target pickleball while moving closer to the soft toy hippo. Bottom row: the robot rotates its gripper to avoid contact with tall bottles and reaches the target matcha can.

C. Physical Experiments

We evaluate IMPACT on a real robotic system consisting of a Franka Panda arm with two Intel RealSense D435 cameras. We use Grounded SAM 2 [15], [21] to generate segmented point clouds for the cost map. We test across 10 real-world scenes with 200 total trials.

Table III shows real world results. IMPACT outperforms LAPP on average across all scenes, with a larger gap on unseen objects (61% versus 40% success), demonstrating strong zero-shot generalization. A user study with the same 25 participants on 10 real-world trajectory pairs also shows IMPACT is preferred in more scenarios. Fig. 3 shows two examples of successful trajectories.

VI. Conclusion

In this work, we introduce IMPACT, a framework for motion planning in clutter that leverages VLMs to assess object contact tolerance, represented in an anisotropic cost map. Our results demonstrate that robots can efficiently reach targets while making semantically-acceptable contact when needed. Current limitations include open-loop execution and reliance on complete RGBD observations; a closed-loop procedure with active perception may address these.

Acknowledgments

We thank our colleagues Ebonee Davis, Sicheng He, Ayano Hiranaka, Minjune Hwang, Yunshuang Li, and Qian (Peter) Wang for helpful technical advice and discussions.

REFERENCES

[1] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[2] R. Tedrake, *Robotic Manipulation*, 2025. [Online]. Available: http://manipulation.mit.edu

[3] M. R. Dogar and S. S. Srinivasa, "A Framework for Push-grasping in Clutter," in *Robotics: Science and Systems (RSS)*, 2011.

[4] OpenAI, A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, A. Madry, A. Baker-Whitcomb, A. Beutel, A. Borzunov, A. Carney, A. Chow, A. Kirillov, A. Nichol, A. Paino, A. Renzin, A. T. Passos, A. Kirillov, A. Christakis, A. Conneau, A. Kamali, A. Jabri, A. Moyer, A. Tam, A. Crookes, A. Tootoochian, A. Tootoonchian, A. Kumar, A. Vallone, A. Karpathy, A. Braunstein, A. Cann, A. Codispoti, A. Galu, A. Kondrich, A. Tulloch, A. Mishchenko, A. Baek, A. Jiang, A. Pelisse, A. Woodford, A. Gosalia, A. Dhar, A. Pantuliano, A. Nayak, A. Oliver, B. Zoph, B. Ghorbani, B. Leimberger, B. Rossen, B. Sokolowsky, B. Wang, B. Zweig, B. Hoover, B. Samic, B. McGrew, B. Spero, B. Giertler, B. Cheng, B. Lightcap, B. Walkin, B. Quinn, B. Guarraci, B. Hsu, B. Kellogg, B. Eastman, C. Lugaresi, C. Wainwright, C. Bassin, C. Hudson, C. Chu, C. Nelson, C. Li, C. J. Shern, C. Conger, C. Barette, C. Voss, C. Ding, C. Lu, C. Zhang, C. Beaumont, C. Hallacy, C. Koch, C. Gibson, C. Kim, C. Choi, C. McLeavey, C. Hesse, C. Fischer, C. Winter, C. Czarnecki, C. Jarvis, C. Wei, C. Koumouzelis, D. Sherburn, D. Kappler, D. Levin, D. Levy, D. Carr, D. Farhi, D. Mely, D. Robinson, D. Sasaki, D. Jin, D. Valladares, D. Tsipras, D. Li, D. P. Nguyen, D. Findlay, E. Oiwoh, E. Wong, E. Asdar, E. Proehl, E. Yang, E. Antonow, E. Kramer, E. Peterson, E. Sigler, E. Wallace, E. Brevdo, E. Mays, F. Khorasani, F. P. Such, F. Raso, F. Zhang, F. von Lohmann, F. Sulit, G. Goh, G. Oden, G. Salmon, G. Starace, G. Brockman, H. Salman, H. Bao, H. Hu, H. Wong, H. Wang, H. Schmidt, H. Whitney, H. Jun, H. Kirchner, H. P. de Oliveira Pinto, H. Ren, H. Chang, H. W. Chung, I. Kivlichan, I. O'Connell, I. O'Connell, I. Osband, I. Silber, I. Sohl, I. Okuyucu, I. Lan, I. Kostrikov, I. Sutskever, I. Kanitscheider, I. Gulrajani, J. Coxon, J. Menick, J. Pachocki, J. Aung, J. Betker, J. Crooks, J. Lennon, J. Kiros, J. Leike, J. Park, J. Kwon, J. Phang, J. Teplitz, J. Wei, J. Wolfe, J. Chen, J. Harris, J. Varavva, J. G. Lee, J. Shieh, J. Lin, J. Yu, J. Weng, J. Tang, J. Yu, J. Jang, J. Q. Candela, J. Beutler, J. Landers, J. Parish, J. Heidecke, J. Schulman, J. Lachman, J. McKay, J. Uesato, J. Ward, J. W. Kim, J. Huizinga, J. Sitkin, J. Kraaijeveld, J. Gross, J. Kaplan, J. Snyder, J. Achiam, J. Jiao, J. Lee, J. Zhuang, J. Harriman, K. Fricke, K. Hayashi, K. Singhal, K. Shi, K. Karthik, K. Wood, K. Rimbach, K. Hsu, K. Nguyen, K. Gu-Lemberg, K. Button, K. Liu, K. Howe, K. Muthukumar, K. Luther, L. Ahmad, L. Kai, L. Itow, L. Workman, L. Pathak, L. Chen, L. Jing, L. Guy, L. Fedus, L. Zhou, L. Mamitsuka, L. Weng, L. McCallum, L. Held, L. Ouyang, L. Feuvrier, L. Zhang, L. Kondraciuk, L. Kaiser, L. Hewitt, L. Metz, L. Doshi, M. Aflak, M. Simens, M. Boyd, M. Thompson, M. Dukhan, M. Chen, M. Gray, M. Hudnall, M. Zhang, M. Aljubeh, M. Litwin, M. Zeng, M. Johnson, M. Shetty, M. Gupta, M. Shah, M. Yatbaz, M. J. Yang, M. Zhong, M. Glaese, M. Chen, M. Janner, M. Lampe, M. Petrov, M. Wu, M. Wang, M. Fradin, M. Pokrass, M. Castro, M. O. T. de Castro, M. Pavlov, M. Brundage, M. Wang, M. Khan, M. Murati, M. Bavarian, M. Lin, M. Yesildal, N. Soto, N. Gimelshein, N. Cone, N. Staudacher, N. Summers, N. LaFontaine, N. Chowdhury, N. Ryder, N. Stathas, N. Turley, N. Tezak, N. Felix, N. Kudige, N. Keskar, N. Deutsch, N. Bundick, N. Puckett, O. Nachum, O. Okelola, O. Boiko, O. Murk, O. Jaffe, O. Watkins, O. Godement, O. Campbell-Moore, P. Chao, P. McMillan, P. Belov, P. Su, P. Bak, P. Bakkum, P. Deng, P. Dolan, P. Hoeschele, P. Welinder, P. Tillet, P. Pronin, P. Tillet, P. Dhariwal, Q. Yuan, R. Dias, R. Lim, R. Arora, R. Troll, R. Lin, R. G. Lopes, R. Puri, R. Miyara, R. Leike, R. Gaubert, R. Zamani, R. Wang, R. Donnelly, R. Honsby, R. Smith, R. Sahai, R. Ramchandani, R. Huet, R. Carmichael, R. Zellers, R. Chen, R. Chen, R. Nigmatullin, R. Cheu, S. Jain, S. Altman, S. Schoenholz, S. Toizer, S. Miserendino, S. Agarwal, S. Culver, S. Ethersmith, S. Gray, S. Grove, S. Metzger, S. Hermani, S. Jain, S. Zhao, S. Wu, S. Jomoto, S. Wu, Shuaiqi, Xia, S. Phene, S. Papay, S. Narayanan, S. Coffey, S. Lee, S. Hall, S. Balaji, T. Broda, T. Stramer, T. Xu, T. Gogineni, T. Christianson, T. Sanders, T. Patwardhan, T. Cunninghman, T. Degry, T. Dimson, T. Raoux, T. Shadwell, T. Zheng, T. Underwood, T. Markov, T. Sherbakov, T. Rubin, T. Stasi, T. Kaftan, T. Heywood, T. Peterson, T. Walters, T. Eloundou, V. Qi, V. Moeller, V. Monaco, V. Kuo, V. Fomenko, W. Chang, W. Zheng, W. Zhou, W. Manassra, W. Sheu, W. Zaremba, Y. Patil, Y. Qian, Y. Kim, Y. Cheng, Y. Zhang, Y. He, Y. Zhang, Y. Jin, Y. Dai, and Y. Malkov, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[5] 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, 1968.

[6] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: Science and Systems (RSS)*, 2013.

[7] L. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1994.

[8] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research (IJRR)*, 2011.

[10] M. Stilman and J. J. Kuffner, "Navigation among movable obstacles: Real-time reasoning in complex environments," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 479–503, 2005.

[11] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," in *Autonomous Robots*, 2016.

[12] H. Nemlekar, Z. Liu, S. Kothawade, S. Niyaz, B. Raghavan, and S. Nikolaidis, "Robotic lime picking by considering leaves as permeable obstacles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[13] A. Xie, Y. Lee, P. Abbeel, and S. James, "Language-conditioned path planning," in *Conference on Robot Learning (CoRL)*, 2023.

[14] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Conference on Robot Learning (CoRL)*, 2023.

[15] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[16] W. Chow, J. Mao, B. Li, D. Seita, V. Guizilini, and Y. Wang, "PhysBench: Benchmarking and Enhancing Vision-Language Models for Physical World Understanding," in *International Conference on Learning Representations (ICLR)*, 2025.

[17] E. Coumans and Y. Bai, "PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning," http://pybullet.org, 2016–2020.

[18] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *IEEE Robotics and Automation Magazine*, 2015.

[19] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3d latents for scalable and versatile 3d generation," *arXiv preprint arXiv:2412.01506*, 2024.

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021.

[21] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.