CAD Assemblies as Object-Level Priors for Robotic Operations Planning

Shruti Chakraborty†, Nathan Damette†, Lucas Labarussiat†, and Caroline Vienne
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{shruti.chakraborty, nathan.damette, lucas.labarussiat, caroline.vienne}@cea.fr

***Abstract*—This paper presents a conceptual framework and proposed architecture for using CAD assemblies as object-level priors for robotic (dis)assembly planning. Rather than reporting experimental results, we contribute (i) a problem formulation, (ii) a descriptor taxonomy for CAD-derived priors (hierarchy, joint relations, hole features, mobility, physical attributes), (iii) a training-free architecture that couples this prior to a skill-grounded Large Language Model (LLM) planner, and (iv) a set of research hypotheses about which descriptors are necessary and sufficient for correct (dis)assembly reasoning. The proposed approach requires no task-specific training, no perception dependency at plan time, and generalises across robot platforms given a pre-defined skill library.**

***Index Terms*—CAD assembly, prior knowledge, robotic disassembly, task planning, scene graph, large language models**

I. Introduction

Robotic (dis)assembly planning is a critical step in automating manufacturing and maintenance tasks, requiring a robot to reason over complex part structures and assemblies. Work on prior maps has shown that a pre-existing structured representation of the robot's surroundings can improve perception and decision-making at deployment. For instance, Zimmerman et al. demonstrated that combining floor plans with semantic cues significantly improves long-term localization [1]. We argue that this idea extends naturally to the object scale: CAD assemblies encode the intra-object structure (parts, joints, fasteners, mobility) that vision-derived inter-object scene graphs cannot recover, and they are routinely produced upstream of the robot's task. Yet existing CAD-based approaches to (dis)assembly planning, such as ASAP [2] or Assemble-Them-All [3], are learning-based pipelines trained on specific CAD datasets, requiring retraining or fine-tuning to adapt to new contexts or unseen assembly configurations.

This paper presents a conceptual proposal for exploiting CAD assemblies as object-scale prior maps to support LLM-based robotic (dis)assembly planning. We introduce a methodological framework that (i) formulates CAD assemblies as structured priors for robotic (dis)assembly; (ii) defines a taxonomy of CAD-derived descriptors relevant to LLM-grounded planning; (iii) proposes a training-free architecture coupling these priors with a skill-grounded LLM planner; and (iv) outlines research hypotheses on the descriptor combinations necessary and sufficient for correct (dis)assembly reasoning.

†Equal contribution.

LLMs represent a promising alternative as task planners: they require no task-specific training, can leverage textual guidelines and maintenance documentation, and may generalise across setups [4]. An LLM-based planner also opens the door to partial planning grounded in specific needs, such as the replacement of a single component, rather than requiring a full disassembly sequence, and can be designed to account for the physical capabilities of the robotic system.

Deploying an LLM as an effective task planner requires grounding it in rich, task-relevant context. SayCan [5] established that LLM planners must be explicitly tied to real-world affordances (the set of actions the robot can execute), rather than sampling plans from the LLM's prior alone. Subsequent work has sought to provide this grounding through structured spatial representations, of which 3D scene graphs [6] have emerged as the dominant format: nodes represent objects or regions, edges carry semantic and spatial relations, and the whole structure can be serialised as natural language. SayPlan [7] showed that hierarchical 3D scene graphs scale LLM planning to building-sized environments by letting the agent collapse and expand regions on demand to stay within its context window. ConceptGraphs [8] demonstrated that such scene graphs can be built online from RGB-D input with open-vocabulary labels, making the prior directly available to a deployed robot.

All of these representations encode *inter-object* relations. For (dis)assembly planning, the LLM instead requires *intra-object* structure: part-level hierarchy, joint types and axes, fastening patterns, mobility constraints.

This information is often not recoverable from perception alone: parts are typically occluded inside assembled objects, and joint kinematics cannot be inferred from external geometry. Perception-based grounding methods such as ReKep [9] and ConceptGraphs [8] themselves note the absence of intra-object structural knowledge (joint axes, fastener relations, internal hierarchy) as a limit of surface-only observation. CAD assemblies, on the other hand, encode this structure by construction.

Recent advances in physics-based assembly sequence planning based on CAD data have yielded promising results. Previous works have looked at extracting part precedence relationships.

In [10], this extraction was done by creating a restriction matrix through collision and gravity analysis using internal

functions available in CAD software. The assembly constraints were then processed with a genetic algorithm to identify the optimal assembly sequence.

JoinABLe [11] proposed to exploit topology and geometry available directly within B-Rep CAD data, and trained a graph neural network architecture for joint axis prediction for each component in a cad assembly.

Assemble them all [3] proposed the assembly by disassembly approach to plan an assembly sequence under physical constraints using individual parts and their assembled states as input. The authors leveraged physics simulation to generate a collision-free assembly path.

This work was extended in ASAP [2], which also incorporated gravitational stability constraints, grasp pose planning and robot arm inverse kinematics, guaranteeing feasible robotic execution of the output assembly plan. The output sequence details each assembly step, including the part to assemble, its motion path, assembly pose, and other parts to be held. Part selection in this method is based on a graph neural network trained on the Fusion 360 Gallery [12] dataset, which calculates a probability distribution over all components in order to estimate the likelihood of each part being next in the disassembly sequence.

These physics- and graph-based methods share two limitations for our target setting. First, they are dataset-specific trained models: adapting them to a new product family or a new CAD format requires retraining or fine-tuning on representative data. Second, their output is a geometric sequence decoupled from the robot's skill repertoire and from natural-language instructions (e.g. maintenance guidelines), which restricts them to full-assembly pipelines and makes targeted operations, such as replacing a single component, difficult to express.

Closest to the approach advocated here, AssemMate [13] recently coupled a graph representation of assembly knowledge to an LLM agent, training a self-supervised graph convolutional encoder to align knowledge-graph entities with the LLM's latent space, and demonstrating assembly-oriented QA and grasping on stacked scenes. Our position differs on three points. (i) *Training-free*: we target a prompt-based pipeline that consumes a serialised CAD-derived graph directly, avoiding a dedicated encoder and the associated training data. (ii) *Explicit CAD-to-prior pipeline*: we specify which descriptors are extracted from the CAD assembly (hierarchy, joint relations, hole features, mobility axes, physical attributes) rather than assuming a pre-existing knowledge graph. (iii) *Skill-grounded, targeted operations*: the LLM planner is prompted with a pre-defined library of robot skills (following AutoRT [14]), so that its output is directly executable and supports partial plans such as single-component replacement. Positioning the CAD assembly as an object-level prior map, rather than as a knowledge graph of abstract assembly facts, is what enables this combination.

II. Proposed solution

Our pipeline (Figure 1) processes CAD data to extract a set of semantic descriptors to describe the CAD assembly architecture in terms of named parts, joint relationships, and physical attributes. This information is used to construct an intermediate semantic representation of the CAD assembly. Given this structured semantic description of the CAD assembly, along with a list of pre-existing robot-executable skills, we develop an LLM-powered assembly planning agent that generates skill-grounded assembly plans for robotic execution.

A. CAD processing and Semantic Information Extraction

The first step of the proposed framework consists in identifying key CAD-derived information useful for generating (dis)assembly plans. Previous works have looked at extracting data such as component level hierarchy [10], part mobility [2], joint relationships [11] between pairs of parts. Component hierarchy indicates ordering constraints within the assembly and informs the LLM how parts are organized into functional sub-assemblies. Joint types (rigid, revolute, planar, cylindrical, ball, screw) between part pairs and joint axes describe the kinematic constraints and degrees of freedom, which would help the LLM determine the type of action needed to separate or connect two parts. Hole features such as hole type (through-hole, counterbore, countersink), diameter, depth, and direction provide information about fastening patterns and insertion axes that helps to determine screwing connections and directly informs tool choice for part-removal. Finally, physical properties like mass and volume for each part can inform sequencing decisions, for instance prioritizing the removal of lighter or more accessible parts first. Establishing which combination of relational, kinematic, and physical descriptors is necessary and sufficient for correct LLM-grounded planning is a central research hypothesis of this work.

We propose to exploit existing CAD-based tools such as Fusion360 [12] for extracting these descriptors. Beyond what the CAD file stores, additional planning-relevant structures like precedence and liaison graphs can be derived automatically from STEP files through collision analysis [15], [16].

B. Generating Intermediate Semantic Representation

Although LLMs reason well over structured semantic information, they are not designed to process raw geometric data [17]. Therefore, once extracted, CAD geometric information needs to be structured. For this, we propose to convert the data to a compressed intermediate representation that can be easily treated by an LLM agent. This is crucial for managing LLM's context window, avoiding context dilution and reducing hallucinations. Several representation formats have been explored in literature, including structured plain text [18], unified code format [19], 3D scene graphs (3DSG) [7], knowledge graphs [20] and latent space graph embedding representations [13].

Hierarchical 3D scene graphs [6] remain one of the most dominant paradigms and most promising solutions for grounding LLM in structured semantic knowlegde. As CAD assem-

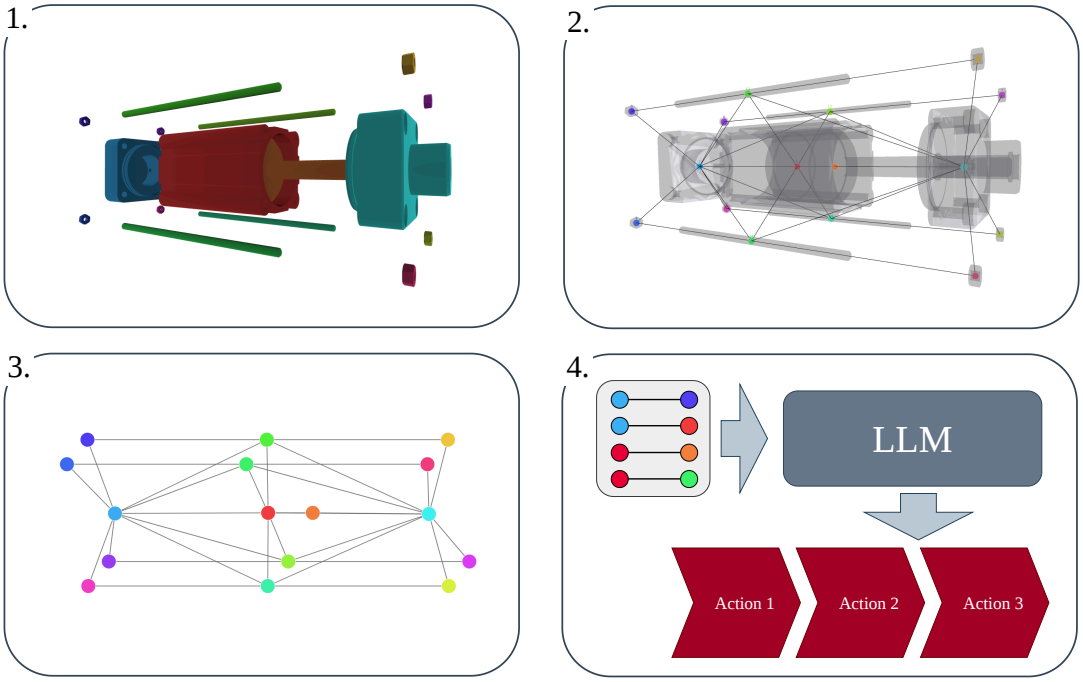

Fig. 1. Proposed pipeline from CAD assembly to robot action plan. (1) The CAD assembly serves as an object-level prior, encoding part structure and spatial organisation. (2) Semantic information is extracted from the assembly geometry and encoded as a graph. (3) The graph is projected into a compact 2D intermediate representation suitable for LLM processing. (4) The serialised representation is provided to the LLM planner, which outputs a skill-grounded action sequence.

blies are inherently hierarchical in structure, this information can easily be encoded as a 3D hierarchical scene graph where nodes would represent parts and edges could contain semantic relationships such as contact points, geometric constraints, joint types and mobility axis. We hence choose to exploit 3DSG as the intermediate representation.

This graph can be passed to the LLM in several ways. The most straightforward is serialisation into JSON format, which can be parsed directly by a pre-trained LLM. For large assemblies, graph-based retrieval techniques can be used to extract sub-graphs of interest before passing them to the LLM [21]. Alternatively, training a graph encoder to compress hierarchical and semantic relationships into a latent space representation has been explored in 3DGraphLLM [22].

C. Skill Grounded Assembly Planning

Using the CAD-derived semantic description of the assembly components, the LLM agent must generate a list of tasks to achieve a target assembly. For each component of the target assembly, the plan is usually composed of a set of pick and place steps to position, align and attach parts together. At a lower level, each step typically maps to a sequence of atomic robot skills (motion, grasp, force control) encapsulated in the skill library.

To create physically feasible plans, the LLM must therefore be prompted with a pre-defined list of robot capabilities. This approach is followed in AutoRT [14], where the LLM task orchestrator is prompted with a list of robot embodiment related limits as well as list of currently available pre-trained robot policies with text summaries of what each collected policy can do. At the task generation step, the LLM is then prompted to select among existing available policies to generate realistic robot task plans.

We adopt a similar approach. We assume the availability of a pre-existing library of robot skills for assembly operations. In our framework a robot ”skill” encapsulates low-level, robot-executable functions. An atomic skill might involve robot trajectory planning or gripper control, whereas a composite skill represents a high-level function, ”pick” or ”place”, that encapsulates multiple low-level atomic skills which includes trajectory planning, grasp planning, and gripper control. [23]

Given a list of skills and the semantic representation of the CAD assembly extracted in the previous section II-B, the LLM planner is prompted to create a structured plan composed of action-level skills chosen from the skill library to achieve the target assembly. Therefore, our method generalises to any robot hardware, provided that a pre-defined list of robot skills is available for that hardware.

III. Discussion and Limits

The proposed framework inherits limits from both of its components. On the LLM side, the planner provides no formal guarantee of optimality: the generated sequence is executable by construction (skill-grounded) but may not minimise cycle time, tool changes, or motion length.

On the CAD side, a risk of discrepancy exists between the as-designed assembly and the as-built object the robot faces:

manufacturing tolerances, wear, prior repairs, or missing parts may deviate from the CAD model. While the prior anchors the plan structure, perception-based checks (pose estimation, part-presence checks, contact feedback) should be used to ground each step at execution time and mitigate this gap. The two are complementary: the CAD prior provides the intra-object structure that perception cannot recover, while perception handles geometric uncertainty at runtime.

A second limit concerns CAD format availability. Not all formats expose the descriptors the pipeline requires: legacy STEP files may lack explicit joint definitions, and scanned meshes carry no part hierarchy. In such cases, a preliminary recovery step is needed; existing tools such as JoinABLe [11] and precedence-graph generators [15], [16] already provide partial solutions that the pipeline can leverage.

Finally, the descriptor taxonomy proposed in § II-A is, at this stage, an informed hypothesis rather than an empirically validated result. Our current intuition, based on the literature surveyed and on preliminary qualitative inspection, is that part hierarchy, a precedence graph, part names, and joint descriptions are necessary for an LLM to produce correct (dis)assembly plans; which additional descriptors (hole features, mobility axes, physical attributes) are also *sufficient* is an open question. Establishing this empirically is part of our planned future work.

A natural next step is an input-descriptor ablation study: comparing LLM-generated plans with and without each class of CAD-derived prior on an assembly benchmark such as Fusion 360 Gallery [12] would quantify the marginal contribution of each descriptor and empirically test the sufficiency hypothesis above.

IV. Conclusion

We have argued that CAD assemblies should be treated as *object-level prior maps* for robotic (dis)assembly. They are the object-scale counterpart to the environment-scale priors (BIM, floor plans, remote-sensing imagery) that have proven useful elsewhere in robot autonomy. The value of such a prior lies in the structure it provides, part hierarchy, joints, fasteners, mobility, and not in its geometric fidelity to the physical object, which perception is better placed to verify at runtime.

The paper contributes a problem formulation, a descriptor taxonomy, a training-free architecture coupling this prior to a skill-grounded LLM planner, and a set of hypotheses about which descriptors are necessary and sufficient for correct (dis)assembly reasoning. The natural next step is to validate those hypotheses empirically: to quantify how much structure from the CAD prior actually changes planning quality and execution support, by ablating each descriptor class on a representative benchmark. Framing CAD as a prior map, rather than only as a geometric model, bridges the prior-map and LLM-planning literatures and opens a new direction for training-free robotic planning.

V. Acknowledgement

This work was supported by the European project euROBIN under Grant Agreement No 101070596 and benefited from state funding managed by the Agence Nationale de la Recherche under the France 2030 program, under reference ANR-24-PEBA-0003.

References

[1] N. Zimmerman, T. Guadagnino, X. Chen, J. Behley, and C. Stachniss, "Long-term localization using semantic cues in floor plan maps," 2022. [Online]. Available: https://arxiv.org/abs/2210.01456

[2] Y. Tian, K. D. Willis, B. Al Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda *et al.*, "Asap: Automated sequence planning for complex robotic assembly with physical feasibility," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4380–4386.

[3] Y. Tian, J. Xu, Y. Li, J. Luo, S. Sueda, H. Li, K. D. Willis, and W. Matusik, "Assemble them all: Physics-based planning for generalizable assembly by disassembly," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–11, 2022.

[4] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," 2022. [Online]. Available: https://arxiv.org/abs/2201.07207

[5] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*. PMLR, 2023, pp. 287–318.

[6] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5664–5673.

[7] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning," *arXiv preprint arXiv:2307.06135*, 2023.

[8] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5021–5028.

[9] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[10] A. Neb and J. Göke, "Generation of assembly restrictions and evaluation of assembly criteria from 3D assembly models by collision analysis," *Procedia CIRP*, vol. 97, pp. 33–38, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827120314189

[11] K. D. D. Willis, P. K. Jayaraman, H. Chu, Y. Tian, Y. Li, D. Grandi, A. Sanghi, L. Tran, J. G. Lambourne, A. Solar-Lezama, and W. Matusik, "JoinABLe: Learning Bottom-up Assembly of Parametric CAD Joints," Apr. 2022, arXiv:2111.12772 [cs]. [Online]. Available: http://arxiv.org/abs/2111.12772

[12] K. D. D. Willis, Y. Pu, J. Luo, H. Chu, T. Du, J. G. Lambourne, A. Solar-Lezama, and W. Matusik, "Fusion 360 Gallery: A Dataset and Environment for Programmatic CAD Construction from Human Design Sequences," May 2021, arXiv:2010.02392 [cs]. [Online]. Available: http://arxiv.org/abs/2010.02392

[13] Q. Zheng, C. Zhang, Z. Liang, E. Lin, S. Cui, Q. Xie, Z. Xu, and L. Zeng, "Assemmate: Graph-based llm for robotic assembly assistance," *arXiv preprint arXiv:2509.11617*, 2025.

[14] M. Ahn, D. Dwibedi, C. Finn, M. G. Arenas, K. Gopalakrishnan, K. Hausman, B. Ichter, A. Irpan, N. Joshi, R. Julian *et al.*, "Autort: Embodied foundation models for large scale orchestration of robotic agents," *arXiv preprint arXiv:2401.12963*, 2024.

[15] G. Pintzos, C. Triantafyllou, N. Papakostas, D. Mourtzis, and G. Chryssolouris, "Assembly precedence diagram generation through assembly tiers determination," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 10, pp. 1045–1057, 2016.

[16] S. Münker and R. H. Schmitt, "Cad-based and/or graph generation algorithms in (dis) assembly sequence planning of complex products," *Procedia CIRP*, vol. 106, pp. 144–149, 2022.

[17] J. Mendez-Mendez, "A systematic study of large language models for task and motion planning with pddlstream," *arXiv preprint arXiv:2510.00182*, 2025.

[18] J. Li, W. Ma, X. Li, Y. Lou, G. Zhou, and X. Zhou, "Cad-llama: leveraging large language models for computer-aided design parametric 3d model generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 18 563–18 573.

[19] Z. Zhang, S. Sun, W. Wang, D. Cai, and J. Bian, "Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models," *arXiv preprint arXiv:2411.05823*, 2024.

[20] G. H. Nguyen, D. Bessler, S. Stelter, M. Pomarlan, and M. Beetz, "Translating Universal Scene Descriptions into Knowledge Graphs for Robotic Environment," Oct. 2023, arXiv:2310.16737 [cs]. [Online]. Available: http://arxiv.org/abs/2310.16737

[21] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," 2025. [Online]. Available: https://arxiv.org/abs/2404.16130

[22] T. Zemskova and D. Yudin, "3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding," 2025. [Online]. Available: https://arxiv.org/abs/2412.18450

[23] S. Chakraborty, L. Labarussiat, F. Gagey, M. Morelli, R. Lallement, Z. Laloux, F. Keith, F. Mayran de Chamisso, R. Gerin, B. Gradoussoff, F. Gosselin, and C. Vienne, "OSCAR: a valuable asset for cost-effective automation," in *Proceedings of the International Symposium on Low-Cost Digital Solutions for Industrial Automation (LoDiSA)*, Cambridge, UK, 2024. [Online]. Available: https://cea.hal.science/cea-04852481v1