2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

IDfRA: Self-Verification for Iterative Design in Robotic Assembly

Nishka Khendry[1] and Christos Margadji[1] and Sebastian W. Pattinson[1]

***Abstract*— Design for Robotic Assembly (DfRA) remains largely dependent on manual planning and heuristic simulation, limiting scalability and robustness in complex industrial settings. Although large language models (LLMs) show promise for semantic reasoning and task planning, most approaches remain tightly coupled to pre-built simulators that assume an accurate world model. We introduce Iterative Design for Robotic Assembly (IDfRA), a closed-loop framework that combines an LLM for plan generation with a vision–language model (VLM) for execution assessment. Given a target structure and a partial environmental signature, the LLM proposes an assembly plan, the robot executes it once at test time, and the VLM evaluates the resulting state to provide feedback for replanning. Through this iterative planning–execution–verification loop, the system progressively improves semantic fidelity and physical feasibility. Crucially, IDfRA does not require an accurate a priori world model before deployment. Instead, physical constraints are discovered online through interaction, enabling adaptation to under-specified environments. Empirical evaluation demonstrates that IDfRA attains 73.3% top-1 accuracy in semantic recognisability, surpassing the baseline on this metric. Moreover, the resulting assembly plans exhibit robust physical feasibility, achieving an overall 86.9% construction success rate, with design quality improving across iterations, albeit not always monotonically. Pairwise human evaluation further corroborates the advantages of IDfRA relative to alternative approaches. By integrating self-verification with context-aware adaptation, the framework evidences strong potential for deployment in unstructured manufacturing scenarios.**

I. Introduction

Robotic assembly involves using robots in manufacturing to automatically assemble components into finished products. While traditionally limited to repetitive tasks for efficiency [1], manufacturing is shifting toward flexible, adaptive systems that operate in dynamic, unstructured environments [2], [3]. This transition has driven interest in Design for Robotic Assembly (DfRA), which involves concurrent design of both products and the robotic systems that assemble them [4]. However, conventional DfRA remains largely manual and template-based [5]–[7], constrained by vast design spaces and reliance on experience-based heuristics – often resulting in suboptimal outcomes [8].

To automate DfRA there has been focus on Assembly Sequence Planning (ASP), which optimises the order of pre-specified assembly steps. Notable examples include a transformer-based model that derives assembly sequences from a target blueprint [9], and a constraint-based tree search to generate feasible sequences automatically [10]. However, such methods address only a subset of DfRA and lack adaptability to dynamic contexts.

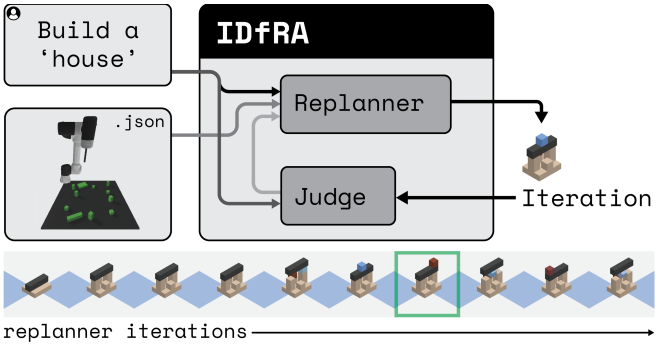

Fig. 1. The overall IDfRA framework. Top: Given a target assembly (e.g., "house") and a signature of the environment in .json format, IDfRA iteratively generates a detailed assembly plan specifying each available block's position and pose. **Bottom**: Renders of the 10 iterations of the "house" assembly.

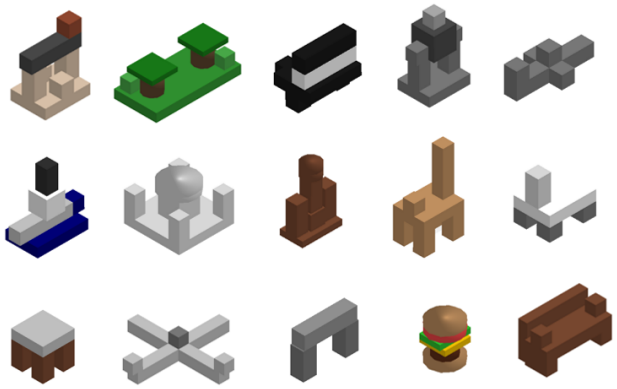

Fig. 2. 15 representative assemblies generated by IDfRA. Top left to bottom right: house, jungle, piano, robot, shark, ship, Taj Mahal, Eiffel Tower, giraffe, sheep, dining table set, ceiling fan, bridge, burger, couch.

Recently, large language and by extension vision-language models (LLMs and VLMs) have shown strong capabilities in semantic reasoning and high-level task planning [11]–[16], including promising results in diverse manufacturing simulations [17]. Leveraging this, [18] proposed the first VLM-driven DfRA system. However, their method selects one from multiple candidate designs generated in parallel, which precludes dynamic, incremental enhancement of design features or learning from past mistakes.

As manufacturing shifts toward flexibility, there is a need for DfRA systems that iteratively improve through feedback. Such adaptability enables learning from previous mistakes and context-aware design refinement, which are crucial in

Codebase: https://github.com/cam-cambridge/open-ended-assembly

This work was supported by the UKRI Engineering and Physical Sciences Research Council award EP/N509620/1.

[1]Institute for Manufacturing, Department of Engineering, University of Cambridge {nk680,cm2161,swp29}@cam.ac.uk

variable industrial environments [19], [20]. To address this, we introduce IDfRA, a framework for Iterative DfRA, which integrates self-verification and re-planning to enable such adaptability; this creates scope to optimally accommodate changing assembly contexts, such as variations in available components, site conditions, or manufacturing tolerances.

IDfRA employs self-verification, a robotic system's ability to assess its own actions and outcomes, as its feedback mechanism. Self-assessment is vital in uncertain or safety-critical environments [21], and has been shown to improve performance by enabling iterative refinement of task plans in both robotics [22]–[24] and other domains [25]. IDfRA addresses three core needs in modern DfRA: *adaptability to dynamic contexts* via iterative refinement, *safety* through embedded self-assessment, and *efficiency* via end-to-end automation. The contributions of this work are:

- IDfRA, which to our knowledge, is the first fully automated framework combining VLM-based self-verification with LLM-based re-planning for iterative assembly design improvement.
- A systematic evaluation showing that IDfRA generates semantically meaningful, physically feasible designs that evolve through feedback, providing insights into foundation model reasoning within DfRA.

II. Related Work

A. Robotic Assembly

Robotic systems are now widely integrated into industrial manufacturing [26], particularly in automotive and aerospace, where they have significantly enhanced production efficiency [2]. Consequently, Design for Robotic Assembly (DfRA) has emerged as a foundational concept.

Early methods for Design for Assembly (later extended to robotics) optimised assembly processes to minimise cost [27]–[29] via rule-based and manual strategies. Their limitations in quantifying executability motivated a shift towards physics-based reasoning, which catalysed widespread adoption of Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) software for visualisation and manual evaluation of components and assemblies [30]–[32]. Contemporary CAD and simulation-based tools, integrated with updated DfRA frameworks derived from the original models [27]–[29] remain the industry standard. Although these approaches advance partial automation of DfRA, they still necessitate close collaboration with engineers throughout the process. Manual DfRA is hindered by a vast solution space, which often leads to suboptimal designs and heavy reliance on designers' procedural knowledge and intuition [8]. Thus, greater automation could significantly improve efficiency and design quality.

Recent shifts in manufacturing towards flexibility and adaptability [33] have amplified interest in automating diverse assembly tasks in unstructured environments [2], [3]. Combined with the limitations of manual methods, this has driven application of ML to robotic assembly, primarily to ASP while DfRA remains largely manual. For example, much ASP research [10], [34]–[37] builds on Assembly-by-Dissaembly (AbD), which analyses disassembly steps to inform assembly plans [38]. Meanwhile, LLMs and VLMs have been employed to decode IKEA instruction manuals [39] and engineering documentation [40], manipulate CAD sketches [41], decompose high-level construction tasks [42], and generate robot code from assembly specifications [43]. These methods demonstrate the significant potential of large pre-trained models in robotic assembly. However, prior ASP work focuses on sequencing predefined designs rather than generating new, physically valid assemblies from scratch.

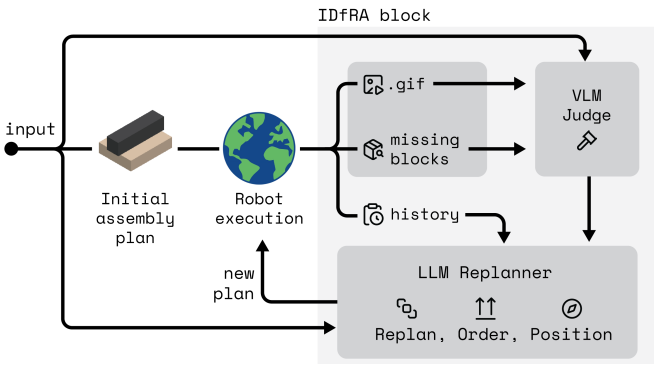

Fig. 3. IDfRA system architecture. A GIF of the initial design execution, any missing blocks, and the original inputs are passed to the *VLM Judge*. Based on the *Judge*'s feedback and previous plans, the *Replanner* produces an updated design which is then executed. This process repeats for 10 iterations and a *VLM Selector* chooses the final assembly.

While BloxNet [18] successfully introduced automated DfRA via VLM-based selection, its architecture has key bottlenecks. Its reliance on physics simulation for internal stability correction limits its real-world applicability, as simulation models often struggle with the inaccuracies of physical assembly [44]. Our framework, conversely, treats simulation only as a final execution placeholder, keeping the reasoning process independent of simulation biases. Furthermore, BloxNet's minimal verification loop (three iterations) and parallel generation strategy preclude the system from learning from previous mistakes. We propose IDfRA, which utilizes an iterative self-verification loop to enable dynamic, context-aware design refinement. Ultimately, we focus on the core challenge identified by [18]: the reliable application of LLMs to embodied robotics.

B. Robotic Planning and Self-Verification with Large Pre-Trained Models

LLMs offer strong potential for robotic planning due to their broad semantic understanding and creativity [45], Yet, they often fail to generate feasible plans in a single pass, and perform far better when combined with external verification and feedback [46]. This insight has driven a surge of research into iterative planning pipelines.

Several works employ execution-derived signals, such as simulator outputs or code errors, as sources of feedback [23], [25], [47]. In contrast, others use language-based correction or guidance to enhance iterative planning such as via

knowledge bases of compliant plans [48] and predefined task guidelines [49]. Furthermore, [24] demonstrates the value of combining diverse feedback sources, such as success detection and scene description, and reports significant improvement in task completion across multiple robot environments. Collectively, these studies demonstrate the value of feedback-driven refinement and the promise of LLMs/VLMs for iterative assembly design.

III. Methods

A. Task Definition

We adopt the Generative DfRA task formulation introduced by BloxNet [18]. Overall system inputs include a target structure name (e.g., "house") and a JSON-formatted list of available blocks (including their dimensions, quantities, and shapes: cuboidal or cylindrical). The goal is to generate a complete assembly plan, specifying for each block its shape, semantic name, colour, 3D centre coordinate (X, Y, Z), and yaw angle.

B. System Architecture

While we adopt the same task definition as BloxNet, our approach addresses its key limitations (Section II-A). Design generation is guided by three objectives: (1) semantic resemblance to the target, (2) physical feasibility for construction by the simulated robot, and (3) iterative enhancement of design features via feedback. The overall architecture and execution flow of IDfRA is detailed in Fig 3. The process begins with a single candidate design generated by BloxNet, with physics-based stability correction disabled to allow refinement of potentially unstable configurations. Object tracking is handled by PyBullet, while missing blocks are identified by comparing the assembly plan against the available set using Python. Assuming that these functions can be replicated in physical systems via vision-based localisation is reasonable–given recent advancements in generalist agents capable of accurate object tracking and pick-and-place [50], [51]. The IDfRA framework consists of four key components:

1) LLM-Based Planning Module (Planner/Replanner): The planning module–referred to as the *Planner* in the first iteration and the *Replanner* in subsequent iterations - shares an identical architecture throughout. It consists of three coordinated LLM instances arranged in series to transform high-level intent into an executable assembly plan as in [18], [23]. It receives the original task specification and, from the second iteration onward, the latest *Judge* feedback together with a record of previous plans. The output is a JSON assembly plan specifying block dimensions and target block poses (3D centre coordinates, and yaw angles), i.e., where each block should be placed.

First, *Plan* generates a high-level design (or in later iterations–referred to as *Replan*–revises the design) while respecting block availability constraints. It specifies each block's semantic role, dimensions, colour (for visualisation), and relative placement. Second, *Order* enforces bottom-up feasibility by correcting illogical construction sequences. Finally, *Position* converts the structured design into explicit spatial coordinates (X, Y, Z coordinates and yaw angles). *Replan* simplifies this step by allowing block dimension switching for reorientation, as in [18], enabling *Position* to focus on spatial layout. This stage required careful prompt engineering to minimise overhangs and enhance structural stability–employing strategies such as removing ambiguity, adding negative examples, and providing detailed instructions that guide positioning based on adjacent blocks. This use of few-shot in-context learning enabled precise placement without relying on a pre-built geometric world model.

2) Robot Assembly Execution Module: The generated plan is executed by a simulated robotic arm in a tabletop environment. Each block is placed via pick-and-place actions with inverse kinematics control. Required reorientations are handled programmatically through 90° roll or pitch adjustments before placement. A suction-based end effector minimises disturbance during stacking.

3) VLM-Based Evaluation Module (Judge): The *Judge*, implemented as a single VLM, evaluates each assembly attempt. It receives a GIF of the executed structure together with structured metadata (block inventory and target specification) describing the generated plan and missing blocks. Using these, the model infers the semantic roles of blocks from visual input and returns structured feedback.

The assessment includes: (1) violations of availability constraints, (2) potential stability concerns, (3) degree of semantic resemblance to the target, and (4) concrete design improvement suggestions. Explicit block availability analysis was included in the prompt to mitigate LLMs' tendency to hallucinate unavailable components – an issue also noted by [18]. Feedback prioritises semantic scene understanding, which current VLMs perform more reliably than detailed physical tracking.

4) VLM-Based Selector Module: After ten refinement iterations, images are rendered for each assembly by spawning blocks at the generated poses under gravity and rigid body dynamics. Designs containing missing components are discarded. The remaining candidates undergo pairwise comparison using a VLM, which selects the more structurally stable and semantically faithful design at each stage. This knockout process continues until a single final assembly is chosen.

C. Implementation Details

The task is executed in a simulated tabletop environment using a Universal Robot UR5e arm and custom suction gripper, both instantiated and controlled using PyBullet. Simulation parameters include: uniform object density 1000 $\mathrm{kg/m^3}$, lateral friction coefficient 0.5, spinning friction coefficient 0.2, gravity -9.81 $\mathrm{m/s^2}$. At each reset, block positions are randomised outside a configurable central region to reserve space for assembly, with minimum spacing to prevent overlap. The simulation acts as an interactive test-time environment: assembly attempts are rendered as GIFs that reflect the realised physical outcome, serving as the core feedback signal for subsequent refinement.

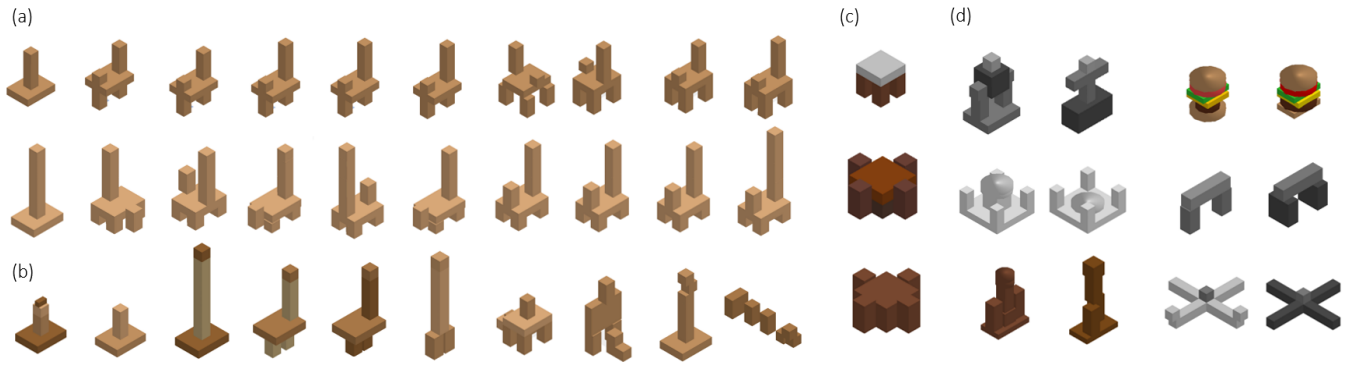

Fig. 4. IDfRA design generation. (a) Multiple runs of IDfRA – each comprising 10 iterations – produce different design evolutions given the same blocks and input "giraffe". (b) 10 candidate designs produced in parallel by BloxNet – each design is produced independently without influence from the previous one. (c) Designs generated by multiple full runs of IDfRA for "dining table set". (d) Designs generated by IDfRA (left) and BloxNet (right) for "robot", "Taj Mahal", "Eiffel Tower", "burger", "ceiling fan", "bridge".

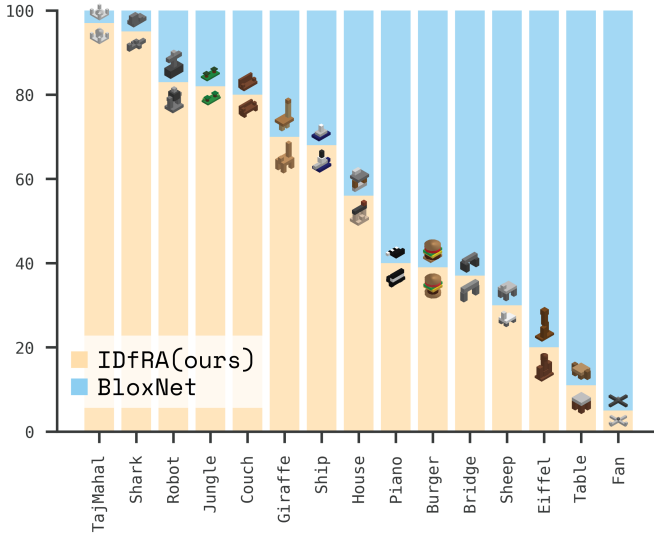

Fig. 5. Human evaluation of final designs. Pairwise voting scores from 100 survey responses comparing IDfRA vs BloxNet across 15 assemblies (best viewed zoomed-in.)

All LLM and VLM instances in this study utilize GPT-4o, accessed via the commercially available API. Consequently, system performance, particularly with respect to outcome assessment, is inherently bounded by the capabilities and limitations of the underlying language model. However, the framework itself is model-agnostic and as foundation models continue to scale and improve, more capable models can be substituted directly without requiring modifications to the overall architecture. JSON is used as the main input at all times, as it is the de facto standard for structured data exchange [52]. Images and GIFs were encoded based on the provided protocols. Temperatures were tuned to balance creativity and consistency at different stages of the pipeline: 0.4 (*Judge*), 0.5 (*Replan*, *Order*), 0.25 (*Position*), and 0 (*Selector*).

IV. Experiments

We evaluate IDfRA on the three objectives defined in Section III-B: semantic recognisability, physical feasibility, and iterative improvement. A representative set of 15 assembly designs was selected to balance structural complexity and tractability: (1) each target structure is sufficiently intricate to support iterative refinement and to incorporate multiple components to achieve semantic resemblance, yet simple enough to be approximated using a limited block set, (2) the structures span varying levels of complexity to illustrate system flexibility.

For each target, a custom block set was prepared to: (1) provide the minimum blocks required for the target structure, (2) include surplus block types to allow creative variation rather than forcing a single design, and (3) remain within the tabletop workspace limits to support physical mappability. All JSON-formatted block sets (including dimensions and quantities) are available in the codebase.

Due to the stochastic nature of autoregressive decoding in VLMs, multiple IDfRA runs produce different design evolutions (Fig. 4a), some outperforming others (Fig. 4c). This variability also highlights IDfRA's ability to adaptively refine plans based on context and feedback. To ensure fair comparability with baseline BloxNet [18], and given the absence of detailed information regarding the exact block sets and number of runs used in its reported results, we ran both methods once using identical inputs (i.e., the target structure name and block set). Each method automatically generated its best designs, and we conducted two quantitative analyses to directly compare their performance. In addition, two analyses were conducted independently on IDfRA's outputs to assess its iterative refinement behaviour and physical feasibility of the generated assemblies.

A. Human Evaluation Survey

We conducted a human evaluation survey in which participants compared each pair of designs (IDfRA vs. BloxNet) for the 15 representative assemblies, and selected the one that more closely resembled the target structure. Order within each pair was randomised to mitigate bias. Participants were instructed to consider overall shape and key features, block usage, and stability. Recruitment combined convenience and snowball sampling.

Responses from 100 participants (Fig. 5) show that IDfRA was preferred in 8 out of 15 designs, achieving a win rate

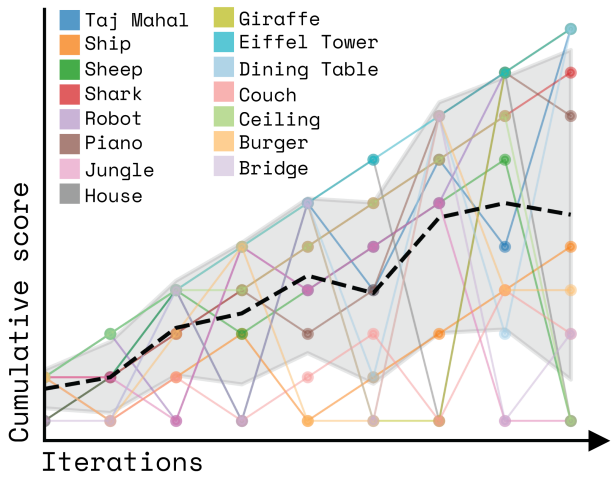

Fig. 6. Iterative design improvement. For each structure, the Y-axis shows the number of times an iteration matched or outperformed previous iterations. The mean trend is indicated by a black dashed line with standard deviation shaded in grey.

of 54.2% (813/1,500 pairwise comparisons). IDfRA outperformed BloxNet on more complex structures with multiple components such as "robot" and "Taj Mahal" (Fig 4d), while BloxNet performed better on simpler targets such as "ceiling fan" and "bridge". In cases like "Eiffel Tower" and "burger", participants favoured BloxNet's designs despite reduced stability and suboptimal block usage, suggesting that they prioritised overall shape over both of these. As in Fig 4d, BloxNet's "Eiffel Tower" was taller but less stable, and its "burger" achieved a flatter profile using a square block for the bun. These results reveal scope for adapting IDfRA's design priorities in future work to reflect the dominant objectives of a given task, whether that be semantic resemblance, structural soundness, or accurate use of materials.

B. VLM Test for Design Recognisability

The second comparative assessment involved prompting a VLM with a rendered assembly image and an unordered list of objects, asking it to rank the objects based on semantic resemblance to the image. This approach, inspired by VLM-based answer scoring methods [53] and previously employed by [18], was reproduced here to enable direct comparability with BloxNet. To avoid trivialising the task with unrelated distractors, we prompted the VLM to generate a list of 200 random objects including and semantically related to the 15 representative assemblies. For each of the 15 assemblies (IDfRA and BloxNet), we constructed ranking trials with list lengths $N = 5, 10, 15, 20$, where each unordered list contained the correct label and $N-1$ distractors. The VLM was then prompted to rank the objects by similarity to the rendered assembly image. Full object lists and code are included in the codebase.

As in [18], we report (1) Top-1 accuracy, the percentage of assemblies in which the correct label is ranked first, (2) Average Ranking, the mean ranking assigned to the correct label across all 15 assemblies, and (3) Relative Ranking, the mean rank of the correct label normalised by the total number of choices N in the list.

N	Method	Top-1 Acc.	Average Ranking	Relative Ranking
5	Iterative DfRA	**64.44%**	**1.60**	**32.00%**
	BloxNet	62.22%	1.62	32.44%
10	Iterative DfRA	**73.33%**	**2.24**	**22.44%**
	BloxNet	57.78%	2.38	23.78%
15	Iterative DfRA	**68.89%**	**2.71**	**18.07%**
	BloxNet	51.11%	3.36	22.37%
20	Iterative DfRA	**46.67%**	**4.38**	**21.89%**
	BloxNet	42.22%	5.40	27.00%

TABLE I. VLM-based design recognisability: Top-1 accuracy, average ranking, and relative ranking averaged across three runs. Computed using VLM rankings of N objects chosen from a pool of 200 objects based on semantic resemblance.

Assembly Design	% Blocks Correctly Placed	% Assemblies Successful
Giraffe	97.78	80
Taj Mahal	100	100
House	100	100
Shark	95	70
Burger	100	100
Jungle	95.71	70

TABLE II. Physical feasibility analysis: Percentage of blocks correctly placed by the robot and percentage of successful assemblies across 10 trials for six assembly designs.

The results of this experiment are presented in Table I. IDfRA consistently outperforms BloxNet in this evaluation, obtaining up to 73.33% top-1 accuracy at $N = 10$. While IDfRA designs generally received higher rankings, both methods exhibited a common limitation: structures with similar high-level shapes were frequently confused – "sheep" and "house" were often misclassified as "table-like" structures such as "side table." Additionally, certain structures disproportionately impacted performance. For example, "shark" and "piano" reduced accuracy for both methods, whereas "sheep" particularly reduced IDfRA's scores, while "Taj Mahal" degraded BloxNet's.

C. Robot Assembly Test for Feasibility

Next, we independently assessed the physical feasibility of IDfRA's generated designs within the simulation environment. For this analysis, we chose six assemblies, as in [18], to cover a range of block types, stacking heights, and difficulty levels. For each, the generated plans were executed in simulation over 10 trials, with the percentage of correctly placed blocks and fully successful build attempts reported in Table II.

This evaluation aims to verify whether IDfRA's designs are in principle constructible, despite using a minimal simulation environment; the setup prioritises design validation over precise robotic optimisation. The minimal simulation serves as a proxy for an industrial system, particularly in providing visual inputs to the *Judge*, and offers a practical platform for proof of concept.

As shown in Table II, IDfRA's designs are physically executable by the simulated robotic arm, with three out of six structures successfully assembled in all 10 trials;

	VLM Recognisability (N = 10)		Containing Missing Blocks	
	Top-1 Acc.	Average Rank	# targets	# iterations
Proposed IDfRA	**94.44%**	**1.05**	**2**	10
PNG instead of GIF	72.22%	1.61	**2**	9
No missing blocks	66.67%	2.22	3	11
No history	83.33%	2.05	4	**5**

TABLE III. Ablation studies: VLM-based design recognisability and frequency of missing blocks across 10 iterations for each of the 6 representative targets ("bridge", "burger", "ceiling fan", "couch", "giraffe", "robot").

the remaining achieving at least 70% success. Failures in "giraffe" were attributed to the lack of collision checks, which caused the tail block to knock over the neck despite the design itself being valid (as evidenced by perfect execution in the other trials). "Shark" and "jungle" exhibited unsuccessful trials due to isolated block placement errors, cuboids at the base edges requiring precise placement sometimes toppled during execution.

Four generated designs spanning core manipulation challenges (e.g., stacking, edge placement), "sheep", "jungle", "burger", and "Taj Mahal", were also successfully assembled in a simple physical environment without plan modification, as shown in Fig 8.

D. Iterative Improvement Evaluation

We examined how assembly designs evolved over ten iterations for each of the 15 target structures. For a given structure, each iteration was compared pairwise against all previous ones (e.g., iteration 3 vs. 2 and 1), yielding a binary outcome: 1 if the later design was better or equal, 0 otherwise. Each comparison was evaluated manually due to empirical limitations of VLMs in highly precise physical reasoning. If one design had missing blocks, it automatically lost. If neither (or both) designs had missing blocks, the more stable design was preferred. For equally stable designs, the more semantically recognisable design won.

For each iteration i, we computed a cumulative score encoding the number of preceding iterations it matched or outperformed. These scores are plotted on the Y-axis of Fig. 6 for iterations 1–9 (iteration 0 being the initial assembly). In this setup, a monotonically increasing line indicates a design that remains the same or improves over time.

Fig 6 illustrates that designs remained the same or improved across iterations on average (black dashed line) with a greater standard deviation (shaded grey) in later iterations. Drops in the coloured lines correspond to unstable or incomplete assemblies, which received lower scores. Overall, the results confirm that designs remain the same or improve via iterative refinement, albeit not always monotonically.

E. Ablation Studies

We conducted three ablations: (1) using a PNG instead of GIF as visual input to the *Judge*, (2) omitting missing block information from the *Judge*, and (3) omitting plan history from the *Replanner*. We analysed designs for 6 representative targets of varying complexity–"bridge", "burger", "ceiling fan", "couch", "giraffe", "robot". We report VLM-based design recognisability (computed as in Table III) and the number of targets (out of 6), as well as number of iterations (out of 6*10 = 60) containing missing blocks.

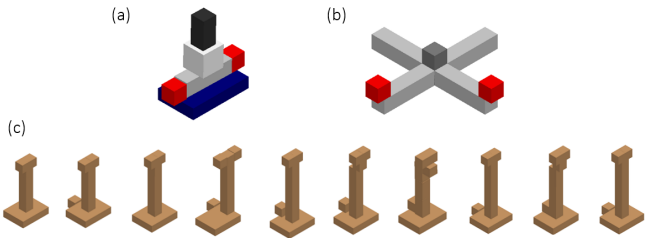

Fig. 7. Limitations. IDfRA occasionally generates designs which contain overhanging blocks or include additional unnecessary details (highlighted in red) such as for "ship" (a) and "ceiling fan" (b). Though rare, it is also possible that no iteration produces a 'good' design such as this run of "giraffe" (c). Such failures largely stem from the VLM misidentifying unstable structures as stable.

The proposed pipeline outperforms all ablations on semantic recognisability. While "No history" produces fewer total missing blocks (5 vs. 10), these failures occur across 4 targets compared to only 2 in the proposed method, demonstrating that the proposed approach concentrates failures in inherently challenging designs ("burger", "robot") rather than causing sporadic failures across multiple targets ("bridge", "burger", "ceiling fan", "giraffe").

V. Conclusion

Experimental results demonstrate that IDfRA generates semantically recognisable and feasible assembly plans, and that iterative refinement can improve Design for Robotic Assembly (DfRA). The approach shows promise as a proof of concept, but several limitations leave scope for further exploration. The LLM-based *Position* submodule remains prone to occasional inconsistencies despite prompt-engineering improvements (Fig 7a, b), though overall performance is acceptable. Generated designs depend, to some extent, on the block set provided and are constrained by how well targets can be represented by a limited set of blocks. Future work could thus focus on scaling the system to more complex tasks and supporting larger, more diverse block sets in a real-world setting. The simulation environment was intentionally kept minimal to serve as a proxy for real robotic systems, with recorded GIFs mimicking real-world execution videos used in the planning pipeline. This design choice enables extension to physical robots and specialised manufacturing settings, where visual feedback plays a central role.

Finally, behaviour varies markedly across runs due to the inherent stochasticity of VLM generation. While this variability can produce diverse and creative "good" designs, it also introduces the rare risk that no satisfactory design emerges (Fig 7c). Moreover, iterative refinement does not always yield monotonic improvement. Future work could address this by incorporating persistent memory, such as a "common mistakes to avoid" module or skill library [25], [35], to encourage more consistent progress across iterations.

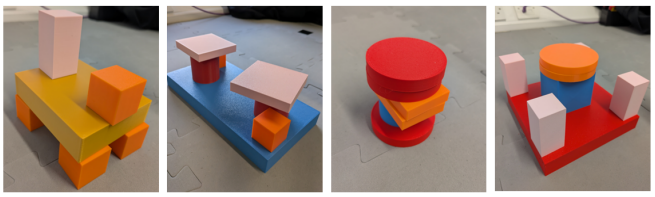

Fig. 8. Physical Experiments. Four generated designs–"sheep", "jungle", "burger", and "Taj Mahal"– assembled using 3D printed blocks by an AGILEX PiPER arm and suction gripper for manipulation, and an Intel RealSense Depth Camera for localisation.

Despite these limitations, IDfRA achieves Top-1 accuracy of up to 73.33% in VLM-based object recognisability, and outperforms BloxNet in 8 out of 15 designs based on pairwise human evaluations – indicating enhanced semantic alignment for complex designs. The generated plans are also physically feasible, with an overall success rate of 86.67% across six representative assemblies. As shown in Fig. 8, the generated designs are directly executable by a real robotic system without plan modification, supporting transferability to real-world settings. Iterative evaluation further shows that designs often improve across successive refinements despite fluctuations. These findings highlight IDfRA's potential as a self-refining, foundation-model-driven system capable of generating executable designs from natural language instructions. IDfRA holds particular promise for flexible robotic assembly in underspecified or evolving contexts, as it can dynamically adapt to feedback and environmental constraints through iterative self-verification and refinement.

ACKNOWLEDGMENT

SWP is a founder of Matta Labs, and CM is an employee; both hold equity in the company, which develops AI systems for manufacturing. The remaining author declares no competing interests.

REFERENCES

[1] J. F. Engelberger and G. C. Devol, "Programmed article transfer," Patent US2 988 237A, May 13, 1954, https://patents.google.com/patent/US2988237A/en.

[2] Y. Jiang, Z. Huang, B. Yang, and W. Yang, "A review of robotic assembly strategies for the full operation procedure: planning, execution and evaluation," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102366, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584522000540

[3] A. R. Das and M. Koskinopoulou, "Toward sustainable manufacturing: A review on innovations in robotic assembly and disassembly," *IEEE Access*, vol. 13, pp. 100 149–100 166, June 2025, publisher's PDF version, copyright ©2025 The Authors.

[4] G. Boothroyd, *Assembly Automation and Product Design*, 2nd ed. CRC Press, 2005. [Online]. Available: https://www.iqytechnicalcollege.com/BAE%20682-Assembly%20Automation%20%26%20Product%20Design.pdf

[5] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics*, vol. 11, no. 1, pp. 137–148, 1969. [Online]. Available: http://www.jstor.org/stable/1266770

[6] T.-C. Chang, R. A. Wysk, and H.-P. Wang, *Computer-Aided Manufacturing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1991.

[7] Z. Bi and X. Wang, *Computer Aided Design and Manufacturing*. Hoboken, NJ: John Wiley & Sons, 2020.

[8] S. Hagemann and R. Stark, "An optimal algorithm for the robotic assembly system design problem: An industrial case study," *CIRP Journal of Manufacturing Science and Technology*, 09 2020.

[9] X. Zhu, D. Jha, D. Romeres, L. Sun, M. Tomizuka, and A. Cherian, "Multi-level reasoning for robotic assembly: From sequence inference to contact selection," 05 2024, pp. 816–823.

[10] Y. Tian, K. Willis, B. Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda, H. Li, S. Chitta, and W. Matusik, "Asap: Automated sequence planning for complex robotic assembly with physical feasibility," 05 2024, pp. 4380–4386.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. 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. M. 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 (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[12] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[13] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as i can, not as i say: Grounding language in robotic affordances," 2022. [Online]. Available: https://arxiv.org/abs/2204.01691

[14] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Gpt-4v(ision) for robotics: Multimodal task planning from human demonstration," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10 567–10 574, 2024.

[15] Y. Jin, D. Li, Y. A, J. Shi, P. Hao, F. Sun, J. Zhang, and B. Fang, "Robotgpt: Robot manipulation learning from chatgpt," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–8, 03 2024.

[16] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," 2025. [Online]. Available: https://arxiv.org/abs/2504.16054

[17] H. Fan, X. Liu, J. Y. H. Fuh, Y. Zhang, and H.-C. Lee, "Embodied intelligence in manufacturing: Leveraging large language models for autonomous industrial robotics," *Journal of Intelligent Manufacturing*, vol. 36, pp. 1141–1157, 2025.

[18] A. Goldberg, K. Kondap, T. Qiu, Z. Ma, L. Fu, J. Kerr, H. Huang, K. Chen, K. Fang, and K. Goldberg, "Blox-net: Generative design-for-robot-assembly using vlm supervision, physics, simulation, and a robot with reset," in *2025 International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[19] K. T. Estelle and B. A. Gozen, "Precision flow rate control during micro-scale material extrusion by iterative learning of pressure-flow rate relationships," *Additive Manufacturing*, vol. 82, p. 104031, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214860424000770

[20] C. Margadji, D. A. Brion, and S. W. Pattinson, "Iterative learning for efficient additive mass production," *Additive Manufacturing*, vol. 89, p. 104271, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214860424003178

[21] A. Aniculaesei, D. Arnsberger, F. Howar, and A. Rausch, "Towards the verification of safety-critical autonomous systems in dynamic environments," *Electronic Proceedings in Theoretical Computer Science*, vol. 232, pp. 79–90, 12 2016.

[22] W. Xia, R. Feng, D. Wang, and D. Hu, "Phoenix: A motion-based self-reflection framework for fine-grained robotic action correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 6981–6990.

[23] F. Joublin, A. Ceravola, P. Smirnov, F. Ocker, J. Deigmoeller, A. Belardinelli, C. Wang, S. Hasler, D. Tanneberg, and M. Gienger, "Copal: Corrective planning of robot actions with large language

models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2024, p. 8664–8670. [Online]. Available: http://dx.doi.org/10.1109/ICRA57147.2024.10610434

[24] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, T. Jackson, N. Brown, L. Luu, S. Levine, K. Hausman, and brian ichter, "Inner monologue: Embodied reasoning through planning with language models," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=3R3Pz5i0tye

[25] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=ehfRiF0R3a

[26] International Federation of Robotics, "World robotics 2024: Industrial robots – executive summary," International Federation of Robotics, Tech. Rep., 2024, accessed: 2025-08-02. [Online]. Available: https://ifr.org/img/worldrobotics/Executive_Summary_WR_2024_Industrial_Robots.pdf

[27] P. Dewhurst and G. Boothroyd, "Early cost estimating in product design," *Journal of Manufacturing Systems*, vol. 7, p. 183–191, 12 1988.

[28] S. Miyakawa and T. Ohashi, "The hitachi assemblability evaluation method (aem)," in *Proceedings of the International Conference on Product Design for Assembly*, Newport, RI, USA, 1986.

[29] L. E. S. Ltd and U. of Hull, *Design For Manufacture and Assembly Practitioners Manual*, version 10 ed. Hull, UK: Lucas Engineering Systems Ltd, 1993.

[30] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics*, vol. 11, no. 1, pp. 137–148, 1969.

[31] T.-C. Chang, R. A. Wysk, and H.-P. Wang, *Computer-aided Manufacturing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1991.

[32] Z. Bi and X. Wang, *Computer Aided Design and Manufacturing*. Hoboken, NJ: John Wiley & Sons, 2020.

[33] P. Kosky, R. Balmer, W. Keat, and G. Wise, "Chapter 10 - manufacturing engineering," in *Exploring Engineering (Third Edition)*, third edition ed., P. Kosky, R. Balmer, W. Keat, and G. Wise, Eds. Boston: Academic Press, 2013, pp. 205–235. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124158917000108

[34] A. Cebulla, T. Asfour, and T. Kröger, "Speeding up assembly sequence planning through learning removability probabilities," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 388–12 394.

[35] Y. Guo, B. Tang, I. Akinola, D. Fox, A. Gupta, and Y. Narang, "SRSA: Skill retrieval and adaptation for robotic assembly tasks," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=RIniw1yin

[36] K. Nagpal and N. Mehr, "Optimal robotic assembly sequence planning (orasp): A sequential decision-making approach," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 9847–9854.

[37] A. Cebulla, T. Asfour, and T. Kröger, "Beyond feasibility: Efficiently planning robotic assembly sequences that minimize assembly path lengths," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 4076–4083.

[38] T.-H. Eng, Z.-K. Ling, W. Olson, and C. McLean, "Feature-based assembly modeling and sequence generation," *Computers & Industrial Engineering*, vol. 36, no. 1, pp. 17–33, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835298001065

[39] C. Tie, S. Sun, J. Zhu, Y. Liu, J. Guo, Y. Hu, H. Chen, J. Chen, R. Wu, and L. Shao, "Manual2skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models," in *The first CVPR workshop on 3D Vision Language Models (VLMs) for Robotics Manipulation: Opportunities and Challenges*, 2025. [Online]. Available: https://openreview.net/forum?id=K3JtdYXT6J

[40] A. C. Doris, D. Grandi, R. Tomich, M. F. Alam, M. Ataei, H. Cheong, and F. Ahmed, "Designqa: A multimodal benchmark for evaluating large language models' understanding of engineering documentation," *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, p. 021009, 12 2024. [Online]. Available: https://doi.org/10.1115/1.4067333

[41] S. Wu, A. Khasahmadi, M. Katz, P. K. Jayaraman, Y. Pu, K. Willis, and B. Liu, "CAD-LLM: Large Language Model for CAD Generation," in *NeurIPS 2023 Workshop on Machine Learning for Creativity and Design*, 2023, outstanding Paper Award.

[42] H. You, Y. Ye, T. Zhou, Q. Zhu, and J. Du, "Robot-enabled construction assembly with automated sequence planning based on chatgpt: Robogpt," *Buildings*, vol. 13, no. 7, 2023. [Online]. Available: https://www.mdpi.com/2075-5309/13/7/1772

[43] A. Macaluso, N. Cote, and S. Chitta, "Toward Automated Programming for Robotic Assembly Using ChatGPT," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 17 687–17 693.

[44] B. Tang, M. Lin, I. Akinola, A. Handa, G. Sukhatme, F. Ramos, D. Fox, and Y. Narang, "Industreal: Transferring contact-rich assembly tasks from simulation to reality," 07 2023.

[45] G. J. Gao, T. Li, J. Shi, Y. Li, Z. Zhang, N. Figueroa, and D. Jayaraman, "VLMgineer: Vision language models as robotic toolsmiths," in *1st Workshop on Robot Hardware-Aware Intelligence*, 2025. [Online]. Available: https://openreview.net/forum?id=i3JNInaLb9

[46] S. Kambhampati, K. Valmeekam, L. Guan, M. Verma, K. Stechly, S. Bhambri, L. P. Saldyt, and A. B. Murthy, "Position: LLMs can't plan, but can help planning in LLM-modulo frameworks," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=Th8JPEmH4z

[47] Y. Jin, D. Li, Y. A, J. Shi, P. Hao, F. Sun, J. Zhang, and B. Fang, "Robotgpt: Robot manipulation learning from chatgpt," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–8, 03 2024.

[48] K. Liang, Z. Zhang, and J. F. Fisac, "Introspective planning: Aligning robots' uncertainty with inherent task ambiguity," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=4TlUE0ufiz

[49] B. Li, P. Wu, P. Abbeel, and J. Malik, "Interactive task planning with language models," *Transactions on Machine Learning Research*, 2025. [Online]. Available: https://openreview.net/forum?id=VmfWywWuYQ

[50] P. Liu, Y. Orru, J. Vakil, C. Paxton, N. Shafiullah, and L. Pinto, "Demonstrating ok-robot: What really matters in integrating open-knowledge models for robotics," 07 2024.

[51] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, and B. Zitkovich, "Rt-1: Robotics transformer for real-world control at scale," 07 2023.

[52] OpenAI, "Introducing structured Outputs in the API," OpenAI Developer Documentation, Aug. 2024, structured Outputs feature for enforced JSON-schema compliance via chat completion API. [Online]. Available: https://platform.openai.com/docs/guides/structured-outputs

[53] L. Fu, G. Datta, H. Huang, W. C.-H. Panitch, J. Drake, J. Ortiz, M. Mukadam, M. Lambeta, R. Calandra, and K. Goldberg, "A touch, vision, and language dataset for multimodal alignment," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.