Learning to Delegate and Act with DELEGACT: Multimodal Language Models for Task-Level Human Cobot Planning in Industrial Assembly

Bram Verstappen
Digital Future Lab
UHasselt - Flanders Make
Diepenbeek, Belgium
bram.verstappen@student.uhasselt.be

Dries Cardinaels
Digital Future Lab
UHasselt - Flanders Make
Diepenbeek, Belgium
dries.cardinaels@uhasselt.be

Danny Leen
Digital Future Lab
UHasselt - Flanders Make
Diepenbeek, Belgium
danny.leen@uhasselt.be

Kris Luyten
Digital Future Lab
UHasselt - Flanders Make
Diepenbeek, Belgium
kris.luyten@uhasselt.be

Raf Ramakers
Digital Future Lab
UHasselt - Flanders Make
Diepenbeek, Belgium
raf.ramakers@uhasselt.be

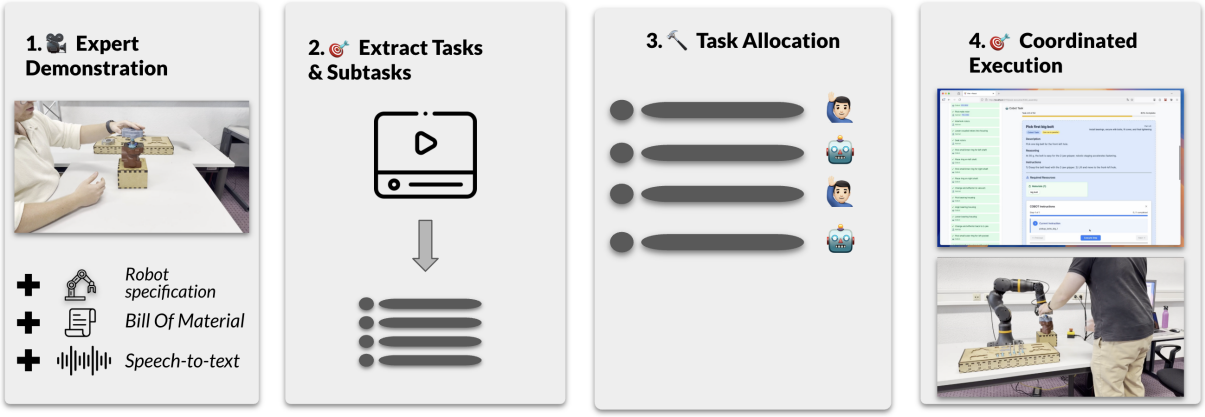


Figure 1: DELEGACT 's human-in-the-loop pipeline for task-level human–robot collaboration. (1) An expert performs a narrated manual assembly. (2) A vision–language model extracts structured steps and atomic subtasks grounded in video frames and transcript. (3) A language model proposes human–cobot task allocations based on robot specifications, operator competencies, and bill-of-material constraints. (4) An interactive interface supports inspection, editing, and coordinated execution with triggerable cobot actions.

Abstract

Industrial assembly is shifting toward human-robot collaboration (HRC) to leverage the complementary strengths of both agents. However, traditional task allocation referred to as the Robotic Assembly Line Balancing Problem (RALBP) remains labor-intensive and often lacks transparency. We introduce DELEGACT, a framework designed to produce workable, intelligible human-cobot task allocations. The framework uses a Vision-Language Model (VLM) to extract atomic operations from expert demonstration videos, then employs a Large Language Model (LLM) to delegate these tasks based on robot specifications, operator competencies, and material definitions. We provide a proof-of-concept prototype and preliminary testing on illustrative cases. Results demonstrate the system's ability to reason about complex constraints such as precision, weight, and ergonomics. This paper illustrates how off-the-shelf foundation models can automate HRC decision-making via a human-in-the-loop paradigm while preserving operator agency and understanding.

CHI EA '26, Barcelona, Spain

ACM ISBN 979-8-4007-2281-3/26/04
https://doi.org/10.1145/3772363.3798803

CCS Concepts

• **Human-centered computing** → **Human computer interaction (HCI)**; **Interactive systems and tools**; **Collaborative interaction**; *Human computer interaction (HCI)*; • **Computing methodologies** → **Multi-agent systems**; **Learning from demonstrations**; • **Computer systems organization** → **Robotics**.

Keywords

Human–robot collaboration, Large language models, Vision Language Models

ACM Reference Format:

Bram Verstappen, Dries Cardinaels, Danny Leen, Kris Luyten, and Raf Ramakers. 2026. Learning to Delegate and Act with DELEGACT: Multimodal Language Models for Task-Level Human Cobot Planning in Industrial Assembly. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3772363.3798803

1 Introduction

Industrial assembly increasingly involves close collaboration between humans and collaborative robots (cobots). Human workers contribute skill and situational awareness, with cobots supporting this work by taking on physically demanding or repetitive actions and executing motions consistently under well-defined conditions [17]. Given these complementary contributions, many manufacturing settings focus less on full automation and more on human–robot collaboration (HRC), in which tasks are distributed to build on the strengths of both partners [14]. In this approach, humans retain oversight and judgment and adapt the process when needed. This perspective aligns with broader shifts toward human-centered automation in Industry 5.0, which prioritize worker well-being and agency while supporting production systems that remain effective as conditions change [3, 13].

A key challenge in HRC is coordinating work at the task level: deciding which steps should be performed by the human or the robot, and in what order. These decisions must account for practical constraints such as safety, tooling, part handling, and the abilities of both agents. In industrial contexts, this problem is often studied through the Robotic Assembly Line Balancing Problem (RALBP) [7, 15]. However, creating effective task allocations remains labor-intensive and highly context dependent [2, 4, 16].

Prior work addresses these challenges through decision-support and planning pipelines. For instance, Gjeldum et al. [12] compare allocations using criteria such as performance and ergonomics, while Chen and Pan [6] generate collaborative plans by modeling feasibility, reachability, and safety. Although these approaches can produce workable allocations, they often require detailed task representations and explicit constraint modeling. Related HCI work such as WeBuild [8] similarly supports delegation in assembly tasks, but assumes that structured step descriptions are provided manually. This motivates methods that can produce initial task lists and delegation plans from higher-level evidence such as demonstrations and natural-language descriptions. At the same time, shifting delegation decisions to automated systems increases the need for transparency and operator control: workers must understand recommendations, intervene when needed [1, 5], and retain the ability to intervene and adjust task allocations through explicit control mechanisms [10].

To lower the effort of producing initial task structures and allocations, recent work uses large language models (LLMs) to generate collaborative plans from high-level inputs. PlanCollabNL [9] derives sub-goals and allocation recommendations from abstract goals and agent conditions, while RoCo [11] refines multi-agent plans through iterative dialogue and environment-based checks. However, much of this work assumes that task steps are already available in a suitable representation for delegation. This suggests a practical next step: converting narrated demonstrations into atomic, delegable task units and producing allocations that operators can inspect, revise, and override during collaboration.

We present DELEGACT, a prototype that aims to lower the barrier to creating task-level human–robot collaboration plans for industrial assembly. Instead of requiring users to author formal task models or detailed constraint specifications, DELEGACT starts from a narrated demonstration and generates a structured task list with a proposed split of responsibilities between a human worker and a cobot. Delegation decisions are informed by readily available contextual inputs, including product information from the bill of materials, robot specifications and limitations, and operator competencies. The system is designed to support intelligibility and user control by exposing intermediate outputs and delegation rationales, enabling operators to inspect, correct, and override decisions before execution. This framing positions automation as an assistive step in collaboration design, supporting faster initial planning while maintaining transparency and agency.

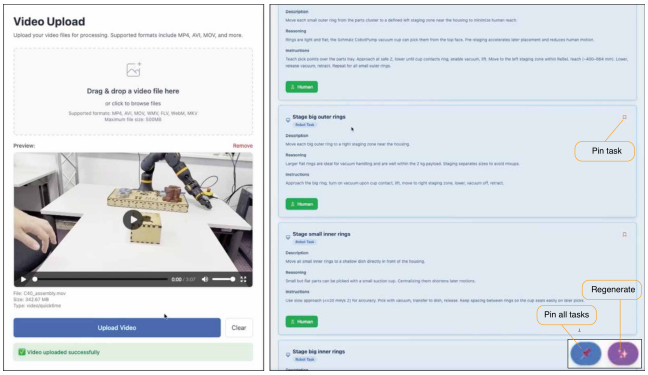
DELEGACT contributes (1) a delegation pipeline that combines VLM-based task extraction from narrated demonstrations with LLM-based human–cobot task splitting, and (2) an interactive prototype that supports inspecting, editing, and regenerating delegation proposals while preserving operator agency.

2 System Overview

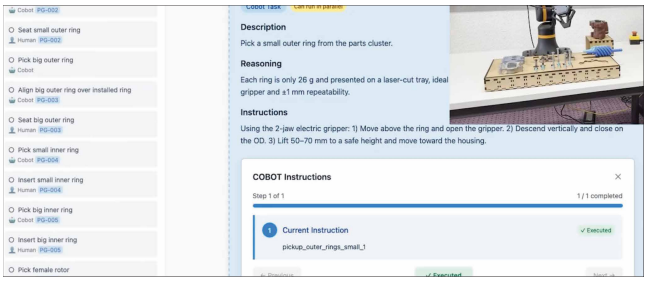
DELEGACT supports task-level human–robot collaboration by turning a narrated expert demonstration into a task assignment plan that users can review, along with step-by-step work instructions. The system takes two main inputs. First, it uses a video of a skilled operator performing the assembly manually while explaining their actions aloud. Second, users provide supporting context about the human worker, the cobot, and the product. This includes operator competencies, robot capabilities and constraints, and a bill of materials with relevant part properties. Figure 2 provides an overview of DELEGACT 's pipeline from the narrated demonstration to task extraction(e.g. Pick up), task assignment, and instruction generation.

Step 1: Atomic task extraction. A VLM/LLM pipeline segments the narrated demonstration into steps and refines them into *atomic tasks* that can be assigned to either the human or the cobot. Atomic tasks are single, self-contained actions, and the system continues splitting steps until this level of granularity is reached. The interface then shows where each step begins and ends so users can review and correct the segmentation. See Figure 2a.

Step 2: Task assignment using human, robot, and product context. The system assigns each atomic task to either the human or the robot. These assignments use the operator's competencies, the robot's capabilities and constraints, and product specifications from the bill of materials, including part size, weight, and material properties. For each task, the system proposes an assignment that fits the human–robot team under these constraints.



(a) Task extraction and delegation review (Steps 1–4).



(b) Execution view with human instructions and cobot actions (Step 5).

Figure 2: Interface walkthrough of DELEGACT. (a) The system ingests a narrated demonstration, extracts atomic tasks, and presents editable human–cobot allocations with reasoning traces. (b) After validation, the system generates step-by-step instructions for both agents and allows trigger-based execution of cobot tasks.

Step 3: Task refinement via splitting and support tasks. The system can refine the workflow by splitting tasks or adding support steps to address limitations of either agent. For example, if bolt tightening requires more precision than the robot supports but repeated tightening is tedious for humans, the system can split the task so the human aligns and lightly tightens the bolt, while the robot performs final fastening.

Step 4: Human-in-the-loop inspection and regeneration. The interface presents the proposed task assignments together with the system's reasoning for why each task is assigned to the human or the cobot. This is shown in Figure 2a. Users can override assignments when they judge that a recommendation does not match the intended workflow or would be better handled by the other agent. After an override, the system updates related tasks and ordering to keep the plan consistent. Users can also pin tasks to preserve selected assignments during regeneration, which helps retain parts of the plan that already match the user's intent when only minor changes are required.

Step 5: Instructions for execution. After verification, the system produces step-by-step instructions for both agents. As shown in Figure 2b. This includes human-readable guidance for operator tasks and task-level robot motions that can be triggered from within the interface. In the current prototype, these robot motions are selected from a predefined library rather than generated through full environment-aware motion planning for the entire assembly.

3 System Architecture

The system is implemented as a web-based interface connected to a backend server (see Figure 3). The backend manages application logic and data processing, and communicates with OpenAI GPT-5 (`GPT5-2025-08-07`) [1] through API calls. The same model is used for both language-only and vision–language inference, with reasoning and verbosity configured to `medium` in both cases. The architecture consists of three main components: a task extraction module, a task delegation module, and an instruction generation module.

Pre Processing. Before the demonstration video can be used by any of the modules, it is first preprocessed. The audio is extracted from the video using FFmpeg [2], after which it is used to generate the operator's think-aloud transcript with OpenAI Whisper [3].

Task Extraction. To extract tasks from a manual assembly demonstration, we process the video using the VLM. For longer recordings, we sample frames every four seconds and combine them into timestamped tiled images, which are provided alongside the operator's think-aloud transcript. A structured task-extraction prompt guides the VLM/LLM to produce a hierarchical procedure description, starting from a small set of contiguous high-level segments and refining these into progressively smaller steps; the full prompt is provided in Appendix A. The model is instructed to further split steps when descriptions contain multiple actions, for example signaled by the word "and," or when a step could plausibly be assigned to both agents, which indicates coarse segmentation. Each extracted step is grounded in both the transcript and the corresponding timestamped frames, and includes execution context such as relevant objects, their properties, and spatial relations.

Task Delegation. Once atomic tasks are available, the LLM assigns each task to either the human or the cobot based on task requirements, robot characteristics, and operator competencies; the full task-delegation prompt is provided in Appendix B. The system first generates an initial assignment on a per-task basis using the task descriptions and transcript, and then performs a second pass to revise the overall sequence and prioritize human ergonomics. The system can also propose workflow reconfigurations to better leverage cobot capabilities and support the operator. These include adding auxiliary support tasks, such as using the cobot as a "third hand," and suggesting alternative execution paths, such as reordering tasks to batch actions before an end-effector change. For tasks assigned to the cobot, the model generates task-specific robot instructions that respect the cobot's constraints, including end-effector movements and tool selection. The system also provides a written reasoning trace for each assignment to support transparency. Finally, it identifies tasks that can be executed concurrently and groups them to support parallel human–cobot execution.

[1] https://platform.openai.com/docs/models/gpt-5
[2] https://www.ffmpeg.org/
[3] https://github.com/openai/whisper

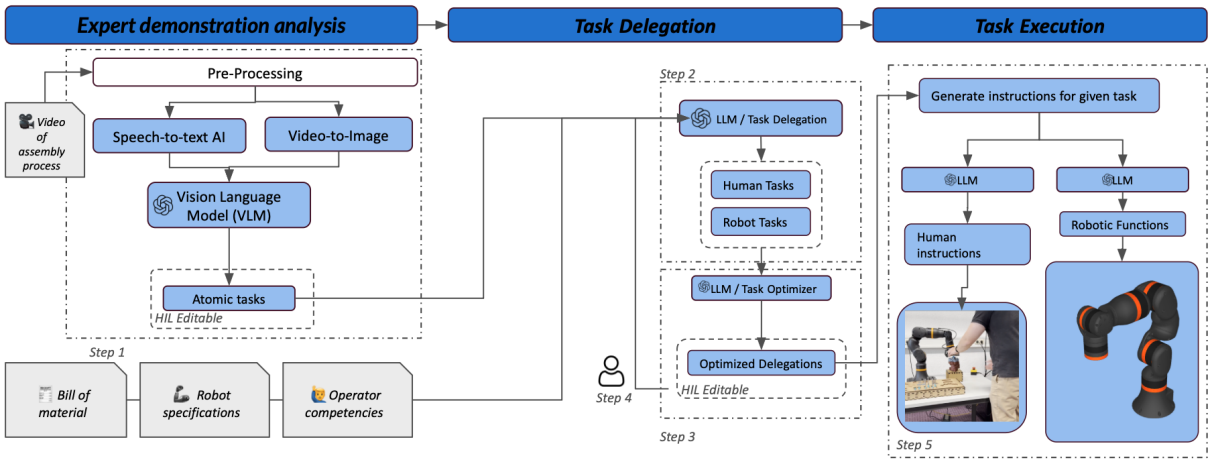


Figure 3: System architecture of DELEGACT. The pipeline consists of three stages: (1) Expert demonstration analysis with speech-to-text transcription and frame sampling for vision–language task extraction; (2) Task delegation using LLM-based human–cobot assignment and workflow optimization; (3) Task execution via parameterized robot behavior templates and generated human instructions.

Task Execution. After task assignment is finalized, the system generates an execution sequence for the cobot tasks using a predefined library of robot behaviors. For each cobot task, the LLM selects and parameterizes the appropriate instruction template, producing a set of machine instructions derived from a human-authored instruction set; the full task-execution prompt is provided in Appendix C. These instructions are presented in the web interface (Figure 2b), which guides the operator through the collaborative assembly process. To support safe use in this proof-of-concept prototype, users must review and approve each cobot instruction before it is executed.

4 Illustrative Cases

We report two illustrative cases (see Figure 4) to demonstrate how DELEGACT supports task extraction and human–cobot task assignment from narrated assembly demonstrations. Both cases were carried out using an Igus ReBeL 6-DOF cobot[4] with a maximum payload of 2 kg, equipped with either a Schmalz CobotPump or a Robolink electric gripper end-effector. These cases are intended to highlight system behavior and interaction patterns rather than to provide a quantitative evaluation.

Assembling a laser-cut VR headset. We first evaluated DELEGACT in a simple assembly task to probe basic task extraction and delegation behavior under minimal complexity (Figure 4a). The case involved assembling a laser-cut virtual reality headset by connecting two plates using finger joints. Since this workflow primarily required precise alignment and insertion, the system assigned the insertion steps to the human operator, reflecting the limitations of the cobot for high-precision manipulation. At the same time, DELEGACT identified opportunities for supportive robot assistance, such as having the cobot pick up parts and hand them to the operator to reduce reaching and improve workflow continuity. The system also proposed a stabilizing support role in a challenging alignment step, assigning the cobot to hold a perpendicular plate in place while the operator aligned and inserted the remaining plate.

Assembling an air compressor. We next applied DELEGACT to a more complex assembly workflow using an air compressor demonstration video (Figure 4b). This case included a broader range of operations, such as tightening bolts, seating bearings, and manipulating parts that exceeded the cobot's payload capacity. The system differentiated between steps that were repetitive versus precision-critical, proposing cobot execution for repetitive actions such as repeated tightening, while assigning fine alignment and insertion steps to the human operator. It also used robot constraints to allocate tasks involving heavy components to the human when the cobot could not safely handle them.

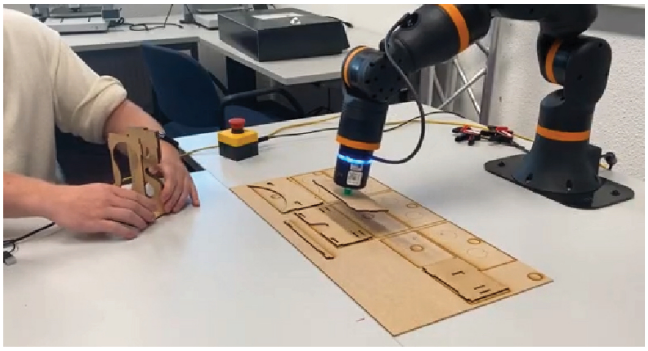
Beyond task assignment, DELEGACT proposed preparatory setup steps to support cobot execution, such as organizing parts in kitting trays and placing components to improve grasping. The system also suggested end-effector swaps when different tools were required. However, in this prototype, it did not account for the time cost of manual tool changes, which reduced the potential benefit of these reconfigurations in practice.

5 Conclusion and Future Work

In this paper, we presented DELEGACT, a proof-of-concept system that supports task-level human–robot collaboration by converting narrated expert demonstrations into reviewable task structures and human–cobot task assignments. By combining VLM-based interpretation of demonstration videos with LLM-based task assignment under user-provided constraints, DELEGACT aims to lower the effort required to produce an initial, editable collaboration plan while preserving operator oversight through an interactive interface.

We demonstrated the system through two illustrative case studies. In both cases, DELEGACT generated task assignments that

[4]https://www.igus.com/product/21465?artNr=REBEL-6DOF-03



(a) Laser-cut VR headset assembly.

(b) Air compressor assembly.

Figure 4: Illustrative case studies used to evaluate DELEGACT. (a) In the VR headset assembly, the cobot assists with part handling and stabilization while the human performs precision alignment and insertion. (b) In the air compressor assembly, the system differentiates repetitive from precision-critical steps, assigns supportive preparation tasks to the cobot, and suggests end-effector changes when required.

reflected practical constraints such as precision requirements and robot limitations, and in some situations proposed supportive collaboration patterns. For example, in the laser-cut VR headset case, the system assigned the cobot a stabilizing "third hand" role to assist with a challenging alignment step. While these cases are not a quantitative evaluation, they highlight the potential of off-the-shelf foundation models to produce delegation proposals that can be inspected, revised, and overridden by operators in collaborative assembly settings.

Building on this prototype, there are several opportunities to strengthen both execution support and interaction in future iterations. Future work will explore tighter integration of robotics-oriented foundation models to reduce reliance on our predefined library of task-level robot behaviors. In particular, platforms such as NVIDIA Cosmos[5] and Gemini Robotics[6] may enable a more dynamic mapping from task descriptions and contextual inputs to executable cobot actions, while keeping the operator in control. In parallel, we plan to integrate state tracking so the interface can monitor assembly progress and automatically advance through the task plan, reducing the need for manual step-by-step confirmation.

[5]https://www.nvidia.com/en-us/ai/cosmos/

[6]https://deepmind.google/models/gemini-robotics/

Acknowledgments

This research was partially supported by Flanders Make, the strategic research center for the manufacturing industry in Flanders through the GenAI - CTO action program (2025-80). This work was funded by the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program, R-13509, and by the Special Research Fund (BOF) of Hasselt University, BOF23OWB29. The infrastructure for this work is funded by the European Union – NextGenerationEU project *MAXVR-INFRA* and the Flemish government. The compressor elements shown in this research prototype are supplied by Atlas Copco.

References

[1] Victoria Alonso and Paloma de la Puente. 2018. System Transparency in Shared Autonomy: A Mini Review. *Frontiers in Neurorobotics* 12 (2018), 83. doi:10.3389/fnbot.2018.00083 eCollection 2018.

[2] Giulia Bassi, Valeria Orso, Silvia Salcuni, and Luciano Gamberini. 2025. Understanding Workers' Well-Being and Cognitive Load in Human-Cobot Collaboration: Systematic Review. *Journal of Medical Internet Research* 27 (2025), e75658. doi:10.2196/75658

[3] Maija Breque, Lars De Nul, and Athanasios Petridis. 2021. *Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry.* Policy brief. European Commission, Directorate-General for Research and Innovation, Luxembourg. doi:10.2777/308407

[4] André Cardoso, Ana Colim, Estela Bicho, Ana Cristina Braga, Débora Pereira, Sérgio Monteiro, Paula Carneiro, Nélson Costa, and Pedro Arezes. 2024. Enhancing Worker Well-Being: A Study on Assistive Assembly to Mitigate Work-Related Musculoskeletal Disorders and Modulate Cobot Assistive Behavior. In *Human Systems Engineering and Design (IHSED 2024)*, Vol. 158. AHFE International, Split, Croatia, 53–62. doi:10.54941/ahfe1005528

[5] Jessie Y. C. Chen, Shan G. Lakhmani, Kimberly Stowers, Anthony R. Selkowitz, Julia L. Wright, and Michael Barnes. 2018. Situation awareness-based agent transparency and human-autonomy teaming effectiveness. *Theoretical Issues in Ergonomics Science* 19, 3 (2018), 259–282. arXiv:https://doi.org/10.1080/1463922X.2017.1315750 doi:10.1080/1463922X.2017.1315750

[6] Qiguang Chen and Ya-Jun Pan. 2024. An Optimal Task Planning and Agent-aware Allocation Algorithm in Collaborative Tasks Combining with PDDL and POPF. arXiv:2407.08534 [eess.SY] https://arxiv.org/abs/2407.08534

[7] Parames Chutima. 2022. A comprehensive review of robotic assembly line balancing problem. *Journal of Intelligent Manufacturing* 33 (2022), 1–34. doi:10.1007/s10845-020-01641-7

[8] C. Ailie Fraser, Tovi Grossman, and George Fitzmaurice. 2017. WeBuild: Automatically Distributing Assembly Tasks Among Collocated Workers to Improve Coordination. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1817–1830. doi:10.1145/3025453.3026036

[9] Silvia Izquierdo-Badiola, Gerard Canal, Carlos Rizzo, and Guillem Alenyà. 2024. PlanCollabNL: Leveraging Large Language Models for Adaptive Plan Generation in Human-Robot Collaboration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Yokohama, Japan, 17344–17350. doi:10.1109/ICRA57147.2024.10610055

[10] Karthik Mahadevan, Mauricio Sousa, Anthony Tang, and Tovi Grossman. 2021. "Grip-that-there": An Investigation of Explicit and Implicit Task Allocation Techniques for Human-Robot Collaboration. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 215, 14 pages. doi:10.1145/3411764.3445355

[11] Zhao Mandi, Shreeya Jain, and Shuran Song. 2024. RoCo: Dialectic Multi-Robot Collaboration with Large Language Models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Yokohama, Japan, 286–299. doi:10.1109/ICRA57147.2024.10610855

[12] M. Crnjac Zizic N. Gjeldum, A. Aljinovic and M. Mladineo. 2022. Collaborative robot task allocation on an assembly line using the decision support system. *INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING* 35 (2022), 510–526. doi:10.1080/0951192X.2021.1946856

[13] Saeid Nahavandi. 2019. Industry 5.0: A Human-Centric Solution. *Sustainability* 11, 16 (2019), 4371. doi:10.3390/su11164371

[14] Amir Nourmohammadi, Masood Fathi, and Amos H.C. Ng. 2022. Balancing and scheduling assembly lines with human-robot collaboration tasks. *Computers & Operations Research* 140 (2022), 105674. doi:10.1016/j.cor.2021.105674

[15] Jacob Rubinovitz, Joseph Bukchin, and Ehud Lenz. 1993. RALB – A Heuristic Algorithm for Design and Balancing of Robotic Assembly Lines. *CIRP Annals* 42, 1 (1993), 497–500. doi:10.1016/S0007-8506(07)62494-9

[16] Christian Weckenborg, Karsten Kieckhäfer, Christoph Müller, Martin Grunewald, and Thomas S. Spengler. 2020. Balancing of assembly lines with collaborative robots. *Business Research* 13, 1 (2020), 93–132. doi:10.1007/s40685-019-0101-y

[17] Guodong Zhang, Xiaowei Luo, Wei Li, Lei Zhang, and Qiming Li. 2025. The Effect of Critical Factors on Team Performance of Human-Robot Collaboration in Construction Projects: A PLS-SEM Approach. *Buildings* 15 (2025), 3685. doi:10.3390/buildings15203685

A Task Extraction Prompt

(1) Identify 3–8 meaningful segments that represent distinct topics or phases in the video. (2) Each segment should be at least 30 seconds long. (3) Look for natural topic transitions, introductions of new concepts, or workflow phases. (4) Create descriptive names that capture the main topic of each segment. (5) Give each segment a list of all the subtasks entailed in this segment. Be detailed, and don't leave out a single subtask. (6) Describe briefly the environment in which each subtask is executed. (7) Be very fine-grained. The subtasks and the environment should be as detailed as possible. (8) Use the given images and their timestamps, in combination with the transcript, to describe the size, characteristics, and relational positioning of the objects to be manipulated in each subtask. Also include this description in the environment section of each subtask. (9) Make sure that the time slots of the segments follow each other—the end time of segment i should always be the start time of segment i+1. (10) Iterate over your decisions; in the last iteration, check all the subtasks in the segments. If there is an "and" in the description of a subtask, split it into two new subtasks. Repeat until no more "and" is found.

B Task Delegation Prompt

You are a master task delegator. Your goal is to classify steps in a production process based on whether they can best be executed by a robotic arm (cobot), a human, or a combination of both a cobot and a human. For each segment to classify, consult the subtasks specified in the segment. For each subtask, determine whether it should be done by the human, the cobot, or both—and explain why the human/cobot should do it and why the cobot/human should not do it. If a segment should be executed by the cobot, describe what the cobot should do in this step while making use of the specifications provided for the robot. Make sure to use only the end effectors given, and limit the robot's interactions to stationary movement of the robot arm. The robot itself must remain in a fixed position. If a segment should be executed by both the cobot and the human, describe what each of them should do in that step. Additionally, provide a clear explanation for why you chose the outcome you did. When classifying a task as cobot or both, you must clarify how the cobot should execute its part of the task using only the provided end effectors, and ensure that the task falls within the capabilities of the robot and end effector specifications.

C Task Execution Prompt

You are a master operator. You operate a cobot arm by programming its execution. You turn written instructions about what the robot should do into executable commands. You will receive an instruction set and may use only the instructions from this set. Additionally, you will receive:
A list of delegations, specifying who should execute each task and how.
A list of segments, to map the delegations to timestamps.
An instruction set, listing all the possible functions you can use to program the cobot arm. The output should be a set of instructions per subtask for a given delegation. Once again, you are not allowed to create new instructions on your own.