Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

Check for updates

Cite this: DOI: 10.1039/d6dd00026f

Constraint-aware labware layout generation from natural language for heterogeneous laboratory robots

Yuya Tahara-Arai, †[ab] Takashi Inagaki, †[c] Akari Kato, [de] Koji Ochiai, [d] Kazuya Azumi, [df] Koichi Takahashi, [df] Genki N. Kanda *[eg] and Haruka Ozaki *[ah]

As laboratory automation accelerates scientific discovery, the manual, expertise-driven task of arranging labware on a robot's deck remains a significant barrier for researchers. This initial setup is crucial for experimental success, safety, and efficiency, yet its complexity grows exponentially with protocol intricacy. To address this challenge, we introduce Labware-Layout Planner, a novel system that integrates a Large Language Model (LLM)-based semantic interpreter with a spatial constraint solver. This architecture translates natural language experimental protocols into optimized, robot-agnostic labware layouts. By interpreting user instructions and physical constraints, our system automates the complex decision-making process of where to place each piece of equipment. We demonstrate its versatility through successful execution of diverse experiments: a liquid handling task and a complex qPCR assay on an Opentrons OT-2, and a multi-step cell passaging protocol on a humanoid Maholo LabDroid. Labware-Layout Planner represents a critical advance by tackling the physical setup phase of automation, paving the way for researchers to move seamlessly from a written idea to robotic execution and freeing human intellect for more creative scientific pursuits.

Received 20th January 2026
Accepted 1st May 2026

DOI: 10.1039/d6dd00026f

rsc.li/digitaldiscovery

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Introduction

Laboratory automation is a promising technology for accelerating scientific inquiry by improving reproducibility and enabling high-throughput experimentation.[1,2] When considering the practical implementation of laboratory automation, there are several areas where humans make implicit decisions. For example, the concept of "Care" proposed by Ochiai *et al.* broadly encompasses tasks traditionally performed implicitly by researchers, such as preparatory work and responses to unexpected events that are necessary to support the execution of an experiment.[3] Among these, a critical preparatory task that has often been overlooked is determining the initial placement of labware. The initial layout in an automated experiment directly impacts the prevention of reagent cross-contamination, the movement efficiency of the robotic arm, and consequently, the overall success and safety of the experiment.[4] However, this task has traditionally relied on the experience and tacit knowledge of researchers.[5] Furthermore, as the variety and number of labware used in a single protocol increase, the number of possible layout patterns increases factorially, making it extremely difficult for humans to determine an optimal solution.[6,7] As a result, labware layout design, which represents implicit human decision-making that is rarely formalized, constitutes a hidden but critical burden in laboratory automation, directly affecting experimental safety, efficiency, and reproducibility. Although recent AI-based systems can generate executable robot protocols, they typically assume that a valid labware layout is manually predefined and do not guarantee satisfaction of physical placement constraints. As highlighted in discussions of hidden human labor in self-driving laboratories,[3] automating this preparatory decision-making step is essential for achieving truly end-to-end laboratory automation.

Laboratory automation workstations are widely used in fields such as drug discovery and biological research.[8,9] Many

[a]*Bioinformatics Laboratory, Institute of Medicine, University of Tsukuba, Tsukuba, Ibaraki, Japan*

[b]*PhD Program in Humanics, School of Integrative and Global Majors, University of Tsukuba, Tsukuba, Ibaraki, Japan*

[c]*Department of Biomedical Engineering, Samueli School of Engineering, University of California, Irvine, CA, USA*

[d]*Laboratory for Biologically Inspired Computing, RIKEN Center for Biosystems Dynamics Research, Kobe, Hyogo, Japan*

[e]*Department of Robotic Science, Medical Research Laboratory, Institute of Integrated Research, Institute of Science Tokyo, Tokyo, Japan. E-mail: kanda.g.0653@m.isct.ac.jp*

[f]*Common Infrastructure and Technology Project, RIKEN Advanced General Intelligence for Science Program, Kobe, Hyogo, Japan*

[g]*Robotics Innovation Center, Research Infrastructure Management Center, Institute of Science Tokyo, Tokyo, Japan*

[h]*Laboratory for AI Biology, RIKEN Center for Biosystems Dynamics Research, Kobe, Hyogo, Japan. E-mail: ai-biology@ml.riken.jp*

† These authors contributed equally.

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

existing automation workstations, such as automated liquid handlers like the Opentrons OT-2, Hamilton STAR, Tecan Fluent, and Beckman Biomek i-Series, as well as dual-arm robots like the Maholo LabDroid, presuppose that reagent containers and tip racks are placed at predefined coordinates.[8–12] While this framework ensures high reproducibility, it leaves challenges in the preparatory phase before the experiment begins.

Our previous study and those of other groups have demonstrated the feasibility of employing large language models to generate robotic scripts for laboratory robots directly from ambiguous natural language instructions.[13,14] Building on this foundation, the present work advances laboratory automation by addressing another fundamental bottleneck: the constraint-aware arrangement of labware. We propose a framework that bridges the gap between unstructured text and physical execution by combining an LLM for semantic extraction with a systematic solver for spatial optimization. Therefore, in this study, we have developed a system that automatically generates a list of necessary items, their initial layout, and simple setup instructions from an experimental protocol written in natural language. Using this system, we conducted an automated colored water dispensing experiment and a PCR mix preparation experiment with an OT-2 (Opentrons Labworks, Inc.), as well as a cell passaging experiment with a general-purpose humanoid robot, the Maholo LabDroid.[15] While we achieved a certain degree of success in these real-world experiments, we also found that the accuracy of some functions has room for improvement. The results suggest the potential to significantly reduce the burden of cumbersome preparatory work, allowing researchers to proceed without having to decide where to place labware, even in an environment like the Maholo with over 180 possible placement locations.

Materials and methods

In this study, we propose a software named "Labware-Layout Planner" that takes a natural language experimental protocol and labware placement constraints as input to generate a protocol for an automated laboratory robot, which includes initial labware placement information (Fig. 1). Based on the content of the natural language protocol and both absolute and relative labware placement constraints, the system outputs a list of necessary items for the experiment and their placement information. The experimental protocol is a text that describes the experimental procedure in natural language. The available labware list specifies the labware available in the target laboratory environment. It is provided as an inventory-based reference to help the model map protocol mentions to canonical labware names and to check basic consistency. It is not treated as a hard gate: mentions that cannot be mapped to the list are flagged for review rather than stopping the pipeline. Placement constraints are the rules that are considered when determining the layout of labware. In this study, we defined two types of constraints: absolute and relative. Absolute constraints describe the relationship between a piece of labware and the equipment, specifically by defining all possible locations where a given labware item can be placed. In contrast, relative constraints describe the relationships between different labware items, for instance, by specifying that certain labware should not be adjacent to others (up, down, left, right, or within the eight surrounding grid cells). These placement constraints must be declared in a JSON file (see SI Appendix). In the current implementation, absolute constraints are provided in a negative form by specifying, for each item, the stations where it cannot be placed together with the total number of stations, and position-planner internally converts these descriptions into the corresponding sets of allowed locations. Note that to simplify the problem, this study only handles relative positions on a grid. These are the potential labware layouts corresponding to the grid of an automation workstation, output by Labware-Layout Planner using the experimental protocol, available labware list, and placement constraints as input. Typically, up to 10 candidates are generated. This is a text that specifies where items should be placed, based on one selected candidate from the labware placement candidates.

Labware-Layout Planner architecture

Labware-Layout Planner consists of three main components: labware-extractor, which determines the list of items; position-planner, which determines the positions of these items based on absolute constraints; and constraint-validator, which filters out any positions that do not satisfy the relative constraints (Fig. 2). Labware-extractor is a program that takes the natural language protocol, available labware list, and placement constraints as input. It interprets the protocol's content and extracts a list of labware to be used from the available labware list (Fig. 2). It utilizes a Large Language Model (LLM), generating the output in JSON format using OpenAI's API (o1-preview; see SI Appendix). During the execution of labware-extractor, if a JSON format error or an OpenAI API error occurs, the system will retry up to 10 times to generate the item list. If an output is generated in the correct JSON format, the process proceeds to the next module, position-planner, regardless of the correctness of the item list itself. position-planner takes the item list and absolute constraints as input and outputs the positions for the items that satisfy these constraints (Fig. 2). By pre-assigning integer values to the candidate placement locations within Labware-Layout Planner, it assigns an integer value to each item in the list obtained from labware-extractor. For the Opentrons OT-2, stations 1 through 11 were pre-assigned. For the Maholo, a total of 189 sequential numbers were pre-assigned to its locations: 7 for 50 mL tube lifters, 22 for 1.5 mL tube lifters, 16 for plate lifters, 120 for CO_2 incubator plate lifters, and 24 for cool incubator 50 mL tube lifters. The module employs a systematic search (brute-force) algorithm to identify valid configurations. To optimize computational efficiency, the solver first assigns positions to items with the most restrictive absolute constraints, effectively pruning the search space before validating the remaining items against relative constraints. Constraint-validator takes the item positions from position-planner and the relative constraints as input, and outputs only the placement information that satisfies these constraints

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

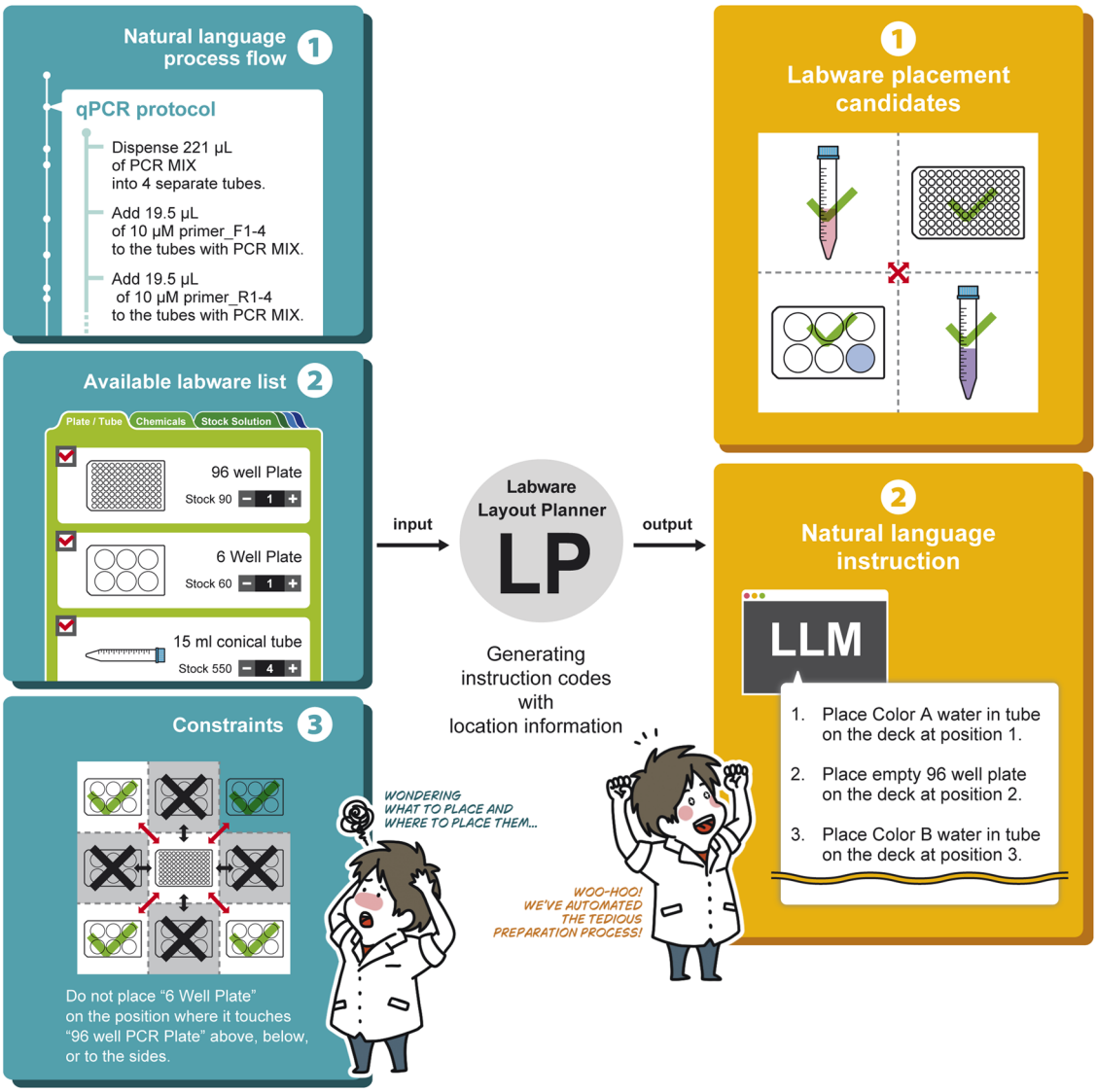

Fig. 1 Overall workflow of the Labware-Layout Planner: from a natural language protocol to an automated labware placement. The Labware-Layout Planner takes a natural language experimental description, an available labware list, and constraints as input. The final output provides a comprehensive labware placement map and a set of natural language instructions for setting up the labware.

(Fig. 2). It ultimately narrows down the options to 10 candidates and outputs one final placement candidate to the user.

Experimental setup

In this study, we used an Opentrons OT-2 (Opentrons Labworks Inc., USA) and a Maholo LabDroid (Robotic Biology Institute Inc., Japan) as automated liquid handlers. The OT-2 was equipped with P20 GEN2 single-channel and P300 GEN2 single-channel pipettes. A Python script was created and executed to operate the OT-2 based on the specified placement information. We generated the OT-2 Python protocol using an LLM with an error-feedback loop. When a generated script raised an error during validation or execution, the error message was provided to the model and the script was regenerated, up to 10 attempts. The Maholo executed jobs generated by ProtocolMaker using its dual arms.

Colored water mixing experiment

Blue and red food coloring (Watashi no Daidokoro Co., Ltd) were prepared in distilled water to create colored solutions. Each solution was dispensed into a 6-well plate. A natural language protocol stating, "mix color A and color B in volume ratios from 300 : 0 to 0 : 300 in 30 µL increments, and dispense 300 µL of each mixture into wells 1–10 of the first column of a 96-well plate," was input into Labware-Layout Planner to automatically generate the placement information (see SI Appendix). An OT-2 script was created based on the generated placement information and executed using the P300_GEN2 pipette. After the operation, the absorbance of each well was measured at wavelengths of 492 nm and 620 nm using a plate reader (Absorbance 96 Plate Reader (Byonoy GmbH, Hamburg, Germany)).

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

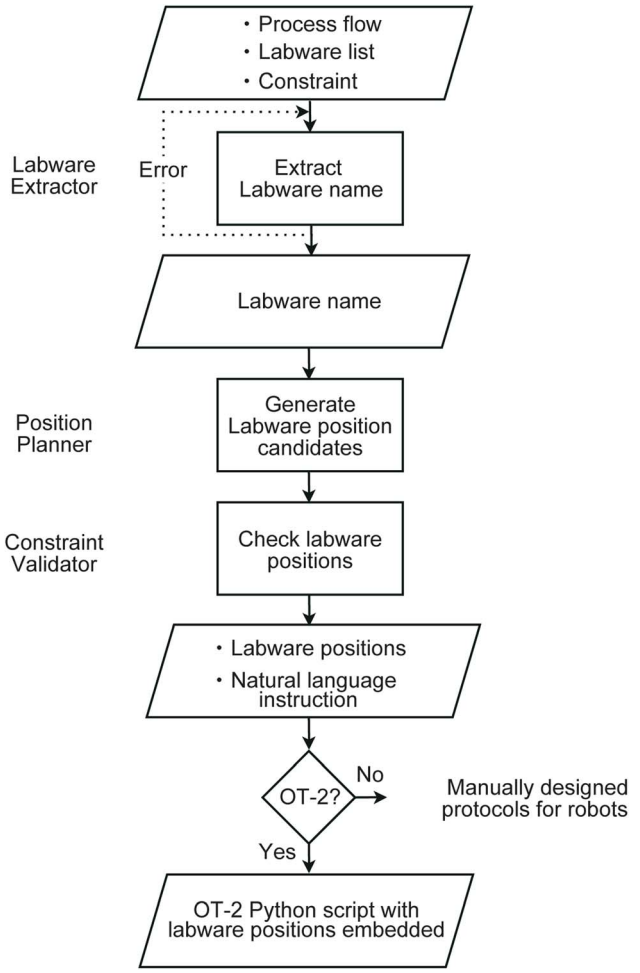

Fig. 2 Architecture of the Labware-Layout Planner system. Workflow for labware placement generation. Inputs include a natural language protocol, an available labware list, and placement constraints. labware-extractor extracts required labware names from the protocol, referencing the available list, with an error handling loop for retries. Position-planner proposes candidate labware positions based on absolute constraints. Constraint-validator filters these positions using relative constraints, yielding the final validated labware positions map.

PCR mix preparation experiment

Reagents and consumables. • qPCR Premix: TB Green Premix Ex Taq™ GC (2×) with ROX Reference Dye II (50×) (Takara, Cat. no. RR071B, Lot no. AN21527A)

• Human Standard cDNA: qPCR Human Reference cDNA, random-primed (50 μg mL^{-1} stock) (Takara, Cat. no. 639654, Lot no. 2104720A)

• Forward/reverse primers (10 μM):[16]

◦ hACTA2_F1/hACTA2_R1

◦ hGAPDH_F1/hGAPDH_R1

◦ hWNT2_F1/hWNT2_R1

• Nuclease-free water: UltraPure™ distilled water (Invitrogen, Cat. no. 10977-015)

• Plates:

◦ StorPlate 96-well round bottom PP, 450 μL (Revvity/PerkinElmer, Cat. no. 6008190) ×2

◦ MicroAmp™ EnduraPlate™ Optical 96-Well Clear Reaction Plate with Barcode (Applied Biosystems/Thermo Fisher Scientific, Cat. no. 4483354)

◦ MicroAmp™ Splash-Free 96-Well Base (Applied Biosystems/Thermo Fisher Scientific, Cat. no. 4312063)

• Tubes:

◦ 1.5 mL tubes (WATSON, Cat. no. 131-815CS-N)

◦ 50 mL conical tubes (FALCON, Cat. no. 352070)

• Pipette tips:

◦ Opentrons OT-2 20 μL tips (Cat. no. 999-00007)

◦ Opentrons OT-2 300 μL tips (Cat. no. 999-00009)

PCR mix prep procedure. A temperature module GEN2 was placed in deck station 9, with its USB cable connected to the rear of the OT-2 and its power cable to an AC outlet. The script was uploaded *via* the Opentrons App v6.0.0, and calibration was performed. The temperature module was pre-cooled to 4 °C from the setup screen. One tube of the 50 μg mL^{-1} human standard cDNA stock was thawed. 3 μL of the stock was added to 47 μL of H_2O in a 50 mL tube, mixed by gentle pipetting, vortexed by tapping, and then briefly centrifuged for 5 seconds (flash spin). The resulting 50 μL of 3 μg mL^{-1} (3000 ng mL^{-1}) cDNA was transferred to a 1.5 mL tube and placed in OT-2 station 1. Two StorPlates were prepared, and 27 μL of H_2O was pre-dispensed into wells A1–A3 of each plate. 3 μL of each forward primer (F1–F3) and reverse primer (R1–R3), taken from 4 °C storage, were added to their respective plates to a final volume of 30 μL per well (1 μM). Each plate was sealed, mixed by tapping, and briefly centrifuged (flash spin). The forward primer plate was placed in station 3, and the reverse primer plate was placed in station 4.

The following reagents were added in order to a 1.5 mL tube on the temperature module:

• 217.6 μL of nuclease-free water

• 12.8 μL of ROX Dye II, completely thawed at room temperature

• 320 μL of TB Green Premix Ex Taq GC

The mixture was pipetted up and down 5 times after each addition, yielding a final volume of 550.4 μL. After preparation, the tube was kept at 4 °C. The script automatically generated by Labware-Layout Planner assembled the reactions in a 96-well PCR plate (station 5) according to the following steps:

1. Dispensed 137.6 μL of the PCR mix into three separate 1.5 mL tubes. To each tube, 3.2 μL of the corresponding forward and reverse primers were added to create three types of primer-specific master mixes (144 μL total each).

2. Dispensed 86 μL of the PCR mix into another tube and added 4 μL of H_2O to create a 90 μL primer-free master mix (for the template-only control).

3. Following the plate map, 18 μL per well of either the primer-specific master mix or the primer-free master mix was dispensed, after which 2 μL per well of template cDNA or H_2O (for no template control (NTC)) was added. A total of 21 wells were used: 3 primer sets × 3 template replicates, 3 primer sets × 3 NTC replicates, and 1 primer-free condition × 3 template replicates.

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

4. After dispensing was complete, the plate was sealed, spun down in a centrifuge at 1000 rpm for 3 min, and checked for bubbles and liquid uniformity.

The sealed plate was placed into a QuantStudio™ 6 Pro (Thermo Fisher Scientific), ensuring correct A1 orientation, and the reaction conditions were set using QuantStudio Design & Analysis Software v2.5.

HEK293A cell passaging experiment

Reagents and consumables. • Dulbecco's Phosphate-Buffered Saline (PBS), 30 mL ×1 (Gibco 10010-023 Lot:2412443)
- Cell culture medium, 30 mL ×1
- FBS (biosera, 555-21245, Lot:015BS427)
- 200 mM L-glutamine (Gibco, 25030081, Lot:2660218)
- Penicillin-Streptomycin (Gibco, 15140-122 Lot:2585661)
- MEM Non-Essential Amino Acids Solution (100×) (Gibco, 11140-050, Lot:2670626)
- DMEM high Glucose (Gibco, 11965-092, Lot:2646159)
- 0.05% Trypsin-EDTA, 30 mL ×1 (Invitrogen, T10282, Lot:2713076)
- 6-Well cell culture plates ×2 (Sumitomo Bakelite, MS-80060)
- 50 mL conical tubes ×3
  - For Maholo: Sumitomo Bakelite (MS-58500)
  - For manual operations: FALCON (352070)
- 10 mL serological pipettes ×3 (Greiner, 607160)
- 1 mL (1000 μL) pipette tips (Sartorius, 791001)
- 200 μL pipette tips (Sartorius, 790201)

Equipment setup. • CO_2 incubator (37 °C, 5% CO_2)
- Aluminum block heater (37 °C, AluminiumBath)
- Cool incubator (4 °C, using rack positions 1–1 to 1–3)
- Plate lifter (position 9)

Reagent warming (day 0–15 min before start). 1. 30 mL of PBS (in a 50 mL tube), 30 mL of medium (in a 50 mL tube), and 30 mL of Trypsin (in a 50 mL tube) were warmed in the aluminum block heater (AluminiumBath) at 37 °C for 15 min.

Cell passaging procedure (day 0). A 6-well plate containing HEK293A cells (2 days post-seeding, seeded at 1×10^6 cells per well) was retrieved from the CO_2 incubator.

The culture medium was aspirated from each well.

2 mL of PBS was added, the plate was gently rocked, and then the PBS was aspirated.

1.5 mL of trypsin was added, and the plate was returned to the incubator for 3 min.

1.5 mL of culture medium was added to neutralize the trypsin, and the cells were fully suspended by pipetting.

2.625 mL of fresh culture medium was pre-dispensed into one well of a new 6-well plate.

375 μL of the cell suspension was added to the well from step (6), and the plate was tilted back and forth and side to side to mix.

The new plate was placed into the CO_2 incubator (37 °C, 5% CO_2) to continue cultivation.

Sample placement. 50 mL tube with PBS to cool incubator rack 1–1

50 mL tube with medium to cool incubator rack 1–2

50 mL tube with Trypsin to cool incubator rack 1–3

Empty 6-well plate to plate lifter 9

6-well plate with HEK293A cells (pre-operation) to CO_2 incubator rack 1–1

Results

We propose Labware-Layout Planner, a software that generates an initial labware layout from protocols and constraints written in natural language.

Colored water experiment using the OT-2

To validate the fundamental performance of Labware-Layout Planner, we used a relatively simple experimental protocol to create a color gradient by mixing colored water (SI Appendix: color water protocol) (Fig. 3A). This protocol was described in natural language and input into the system along with a list of available labware and placement constraints (Fig. 3B). From these inputs, Labware-Layout Planner performed labware name extraction *via* labware-extractor, followed by constraint-aware placement determination using position-planner and constraint-validator, ultimately outputting a labware placement map and natural language setup instructions (Fig. 3B). Furthermore, based on the work of Inagaki *et al.*,[13] we also generated a Python script for the OT-2 using this placement information. As a result of 10 independent trials using the same colored water protocol, the labware name extraction (by labware-extractor) was successful in 9 out of 10 trials.

Labware extraction

As the first step, the necessary labware for the experiment was extracted from the natural language protocol using an LLM (labware-extractor). The LLM was instructed to reference the available labware list (see SI Appendix) and output the results in a specific JSON format. This JSON includes the labware name, the liquid quantity (quantity), the unit of the liquid (unit), the initial content of the labware (init_content), and the labware name and identifier (id) for Opentrons (labware.id, labware.name). In the 10 trials using the colored water protocol, labware-extractor successfully and correctly extracted the necessary labware list 9 times. However, one trial failed at this extraction step.

Constraint validation

Based on the labware list extracted by labware-extractor, position-planner generated placement candidates for the available slots (1–11) on the OT-2 deck. Next, constraint-validator evaluated these candidates and filtered out any layouts that violated predefined absolute constraints (*e.g.*, a specific slot cannot be used) and relative constraints (*e.g.*, "color A water" and "color B water" should not be adjacent, see Fig. 3B input). In all 9 trials where the extraction by labware-extractor was successful, a valid labware layout that satisfied these constraints was determined and output as a placement map (see Fig. 3C for an example layout).

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

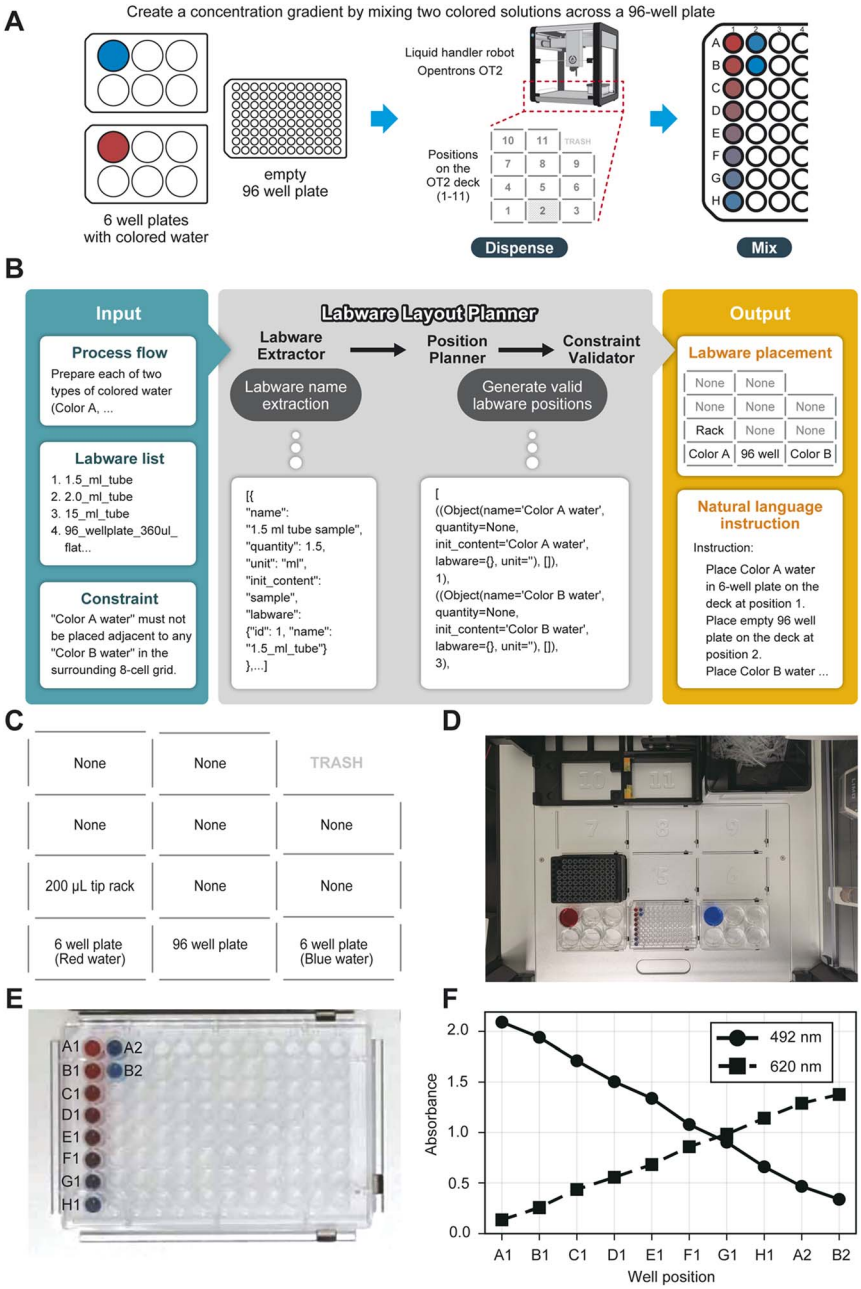

Fig. 3 Colored water experiment: validation of Labware-Layout Planner. (A) Overview of the experimental protocol. Two colored solutions (color A and color B) are initially in separate 6-well plates and are then transferred and mixed in a 96-well plate by a liquid-handling robot Opentrons OT-2 to create a gradient. The OT-2 deck layout with numbered positions is shown. (B) Detailed workflow example using Labware-Layout Planner. Inputs (natural language process flow, available labware list, and placement constraints) are processed through labware-extractor (labware name extraction), position-planner (generation of valid labware positions), and constraint-validator (constraint validation) to produce outputs, a labware placement map and natural-language setup instructions. Examples of actual inputs and intermediate outputs (*e.g.*, JSON) are displayed. (C) The labware placement map and natural-language setup instructions generated by Labware-Layout Planner. (D) A photograph of the actual experimental setup on the Opentrons OT-2 deck layout specified in (C). (E) A photograph of the 96-well plate showing the resulting color gradient after the OT-2 executed the dispensing and mixing protocol. (F) Corresponding absorbance measurements at 492 nm (solid line) and 620 nm (dashed line) across selected wells, confirming the successful creation of the gradient.

OT-2 experimental validation

Using the determined labware placement information and the original experimental protocol as input, a Python script executable on the OT-2 was generated using an LLM. The generated scripts were first checked for basic executability (*e.g.*, absence of syntax errors or incorrect API calls) using the opentrons_simulate tool. All generated Python scripts passed the opentrons_simulate check and were deemed valid and executable.

To confirm the practical utility of the Python script generated by Labware-Layout Planner and verified by the simulation, we

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

conducted a real-world experiment using an actual Opentrons OT-2 robot (Fig. 3D). We executed one of the scripts (one of the nine successful trials) and observed the process as the OT-2 automatically aspirated colored water (color A, color B) from the 6-well plates and dispensed and mixed them in the specified ratios into a 96-well plate. As a result of the experiment, the intended color gradient was formed on the 96-well plate, as shown in Fig. 3E. Furthermore, when we measured the absorbance of each well using a plate reader (Fig. 3F), the absorbance values varied incrementally according to the ratio of color A (blue solution, absorbance peak at 620 nm) and color B (red solution, absorbance peak at 492 nm), confirming that the mixing and dispensing operations were accurately executed based on the protocol generated by Labware-Layout Planner.

qPCR reagent preparation experiment using the OT-2

To validate the practical utility of Labware-Layout Planner on a more complex task, we used a qPCR reagent preparation protocol involving multiple reagents and primer sets (SI Appendix: OT-2 qPCR protocol) (Fig. 4A). In this experiment, an OT-2 robot dispenses and mixes three types of primer sets, sample DNA, PCR mix, and water (NTC) into a 96-well PCR plate. Similar to the colored water experiment, we input the natural language protocol, available labware list, and placement constraints into Labware-Layout Planner (Fig. 4B). The system performed labware name extraction (labware-extractor) and placement determination (position-planner, constraint-validator) to generate a placement map, setup instructions, and an OT-2 Python script.

Labware extraction

In the qPCR experiment, the first step was again labware extraction from the natural language protocol by labware-extractor (Fig. 4B). Using the same procedure as the colored water experiment, the necessary labware and its information were extracted into a JSON format using an LLM. The qPCR protocol is more complex than the colored water experiment, involving a greater variety and number of labware items. As a result of 10 trials, labware-extractor was able to correctly extract the necessary labware list 5 times, while the remaining 5 trials failed at this step. Across ten qPCR reagent-preparation trials, one completed successfully, while nine failed. In five cases, the failures occurred during the early extraction stage after several rounds of corrective looping: as the prompt and intermediate outputs accumulated, the model's context window became insufficient, which led to incomplete or inconsistent extractions and invalid structured outputs. In the remaining four cases, the extraction and layout generation proceeded, but validation failed because the model produced labware names that did not match the canonical Opentrons definitions and were rejected by opentrons_simulate.

Constraint validation

Based on the labware list extracted by labware-extractor, position-planner and constraint-validator determined the placement (see Fig. 4B). While our system can incorporate absolute and relative placement constraints that reflect real-world laboratory conditions, the qPCR experiment required so many labware items that the OT-2 deck was nearly full. Applying ideal constraints (*e.g.*, separating tip racks from reagents) extremely limited the possible placement options. Therefore, we did not impose additional user-defined absolute or relative placement constraints in this experiment beyond OT-2 hardware feasibility and deck capacity. Under this setting, the role of position-planner and constraint-validator was to confirm feasibility and to output one collision-free, hardware-valid placement in which all required labware fit on the OT-2 deck. In all 5 trials where extraction was successful, a labware layout was determined and output as a placement map (example layout: Fig. 4C Output; setup based on an actual layout: Fig. 4D).

OT-2 experimental validation

Based on the determined labware placement information and the original protocol, OT-2 Python scripts were generated using an LLM for each of the 5 successful extraction trials. These scripts were then validated for executability (checking for syntax errors, API call errors, *etc.*) using the opentrons_simulate tool. Only one script passed this validation and was obtained as a viable candidate for the qPCR mix preparation protocol. To confirm the validity of this automatically generated and validated Python script, we performed an actual qPCR reagent preparation experiment. Specifically, we used the script to prepare reaction mixes for three primer sets against a human cDNA sample, including both sample (cDNA) and no template control (NTC, water) reactions in triplicate (21 wells total). First, a human operator placed the PCR mix, primers, sample DNA, and water on the OT-2 deck according to the protocol. The generated script was then executed to start the automated experiment. The OT-2 used p20 and p300 pipettes to automatically perform tasks such as adding and mixing primers into the PCR master mix, and dispensing the template (cDNA or water) into the appropriate wells.

Real-time PCR was performed on the prepared plate using a QuantStudio™ 6 Pro. The results showed that for all three primer sets, the cDNA wells produced the expected amplification curves, while the NTC wells showed no amplification or only a very weak signal at a very late cycle (Table 1). For example, as shown in Fig. 4E, good amplification was observed for targets like hACTA2 and hGAPDH. As shown in Table 1, the mean C_q value for hACTA2 cDNA was 19.273 *versus* 35.411 for the NTC, and for hWNT2 cDNA it was 28.532 *versus* 38.269 for the NTC, confirming specific amplification. Furthermore, no amplification was observed in the hGAPDH NTC or the no-primer cDNA wells, indicating a low risk of contamination. These results demonstrate that the labware layout and accompanying Python script generated by Labware-Layout Planner can produce accurate and reproducible results even in qPCR reagent preparation.

Cell passaging experiment using Maholo

To test the versatility of Labware-Layout Planner on a different robot platform and a more complex biological task, we conducted a HEK293A cell passaging experiment using the dual-

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

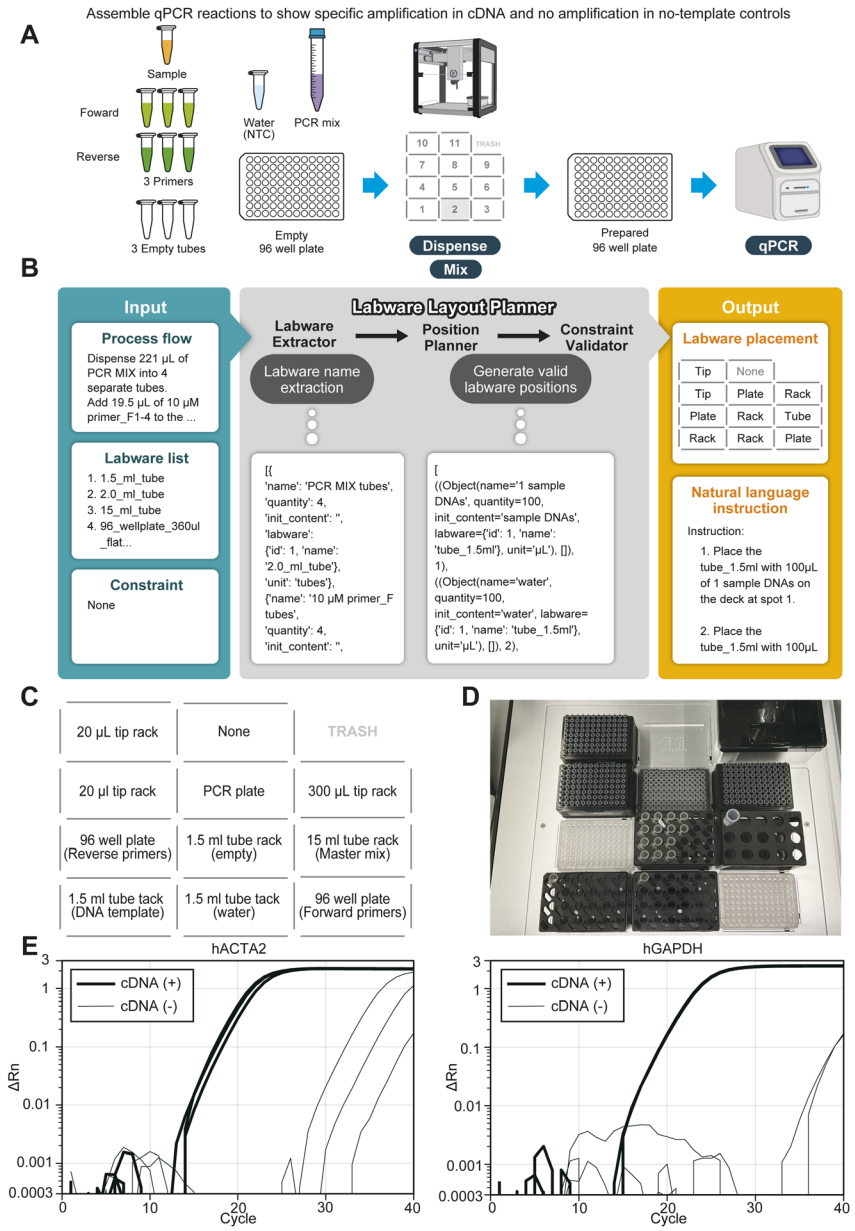

Fig. 4 Validation of Labware-Layout Planner in qPCR reagent preparation using Opentrons OT-2. (A) Overview of the qPCR experimental protocol. Samples, 3 sets of forward and reverse primers, water (no template control – NTC), and PCR master mix are prepared in various source labware. These are then dispensed and mixed into an empty 96-well plate by the Opentrons OT-2. The prepared plate is subsequently analyzed using a qPCR instrument. The OT-2 deck layout is shown. (B) Detailed workflow example for the qPCR experiment using Labware-Layout Planner. Inputs (natural language process flow, available labware list, and placement constraints) are processed through labware-extractor (labware name extraction), position-planner (generation of valid labware positions), and constraint-validator (constraint validation) to produce outputs (a labware placement map and natural-language setup instructions). Examples of actual inputs and intermediate outputs (*e.g.*, JSON) are displayed. (C) The labware placement map and natural-language setup instructions generated by Labware-Layout Planner for the qPCR experiment. (D) A photograph of the actual labware setup on the Opentrons OT-2 deck for the qPCR experiment layout specified in (C). Labware, including 20 µL tip racks, a PCR plate, a 300 µL tip rack, a 96-well plate for reverse primers, 1.5 mL tube racks for DNA template and water, a 15 mL tube for master mix, and a 96-well plate for forward primers were placed according to the labware placement map generated by Labware-Layout Planner (left). (E) Representative qPCR amplification curves (ΔR_n *vs.* Cycle) for two target genes: hACTA2 (left) and hGAPDH (right).

arm robot "Maholo" (Fig. 5). This experiment involves a multi-step process, including media exchange, cell suspension from a cultured 6-well plate, and transfer of the cells to a new 6-well plate (Fig. 5A). We described this cell passaging protocol in natural language and input it into the system along with a list of labware available for Maholo and the relevant placement constraints (Fig. 5B). Based on these inputs, Labware-Layout Planner performed labware name extraction (labware-extractor) and constraint-aware placement determination (position-planner and constraint-validator) to ultimately output

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Table 1 qPCR results from OT-2-based reagent preparation using Labware-Layout Planner. Summary of qPCR amplification outcomes for 3 target genes using cDNA and no template controls (NTCs), following reagent dispensing protocols automatically generated by the Labware-Layout Planner system. Each target was tested in triplicate. Mean quantification cycle (C_q) values and standard deviations (SD) are reported for both cDNA and NTC conditions. The absence of amplification (NaN) in NTC wells indicates low background and minimal contamination

Sample	Target	No. of replicates	C_q mean	C_q SD
NTC	hACTA2	3	35.411	2.219
cDNA	hACTA2	3	19.273	0.469
NTC	hGAPDH	3	NaN	NaN
cDNA	hGAPDH	3	21.135	0.053
NTC	hWNT2	3	38.269	NaN
cDNA	hWNT2	3	28.532	0.085
cDNA	no_primer	3	NaN	NaN

an initial labware placement map for the Maholo workspace and natural language setup instructions.

Labware extraction

As the first step, labware-extractor extracted the necessary labware from the natural language cell passaging protocol. The protocol included items such as 50 mL tubes for PBS, medium, and trypsin, as well as 6-well plates for cell culture. labware-extractor correctly interpreted this information and successfully generated a list in JSON format, including contents and IDs, as shown in the output example (object) in Fig. 5B.

Constraint validation

Next, based on the labware list generated by labware-extractor, position-planner and constraint-validator determined the optimal placement on the Maholo platform. During this process, Maholo-specific absolute placement constraints, such as "tubes must be placed in a specific tube rack," were taken into account. As a result, a viable labware placement candidate was generated.

Maholo experimental validation

Based on the placement information output by Labware-Layout Planner (Fig. 5C), we set up the reagents and labware in the Maholo's workspace (Fig. 5D). We manually created a Maholo motion generation protocol corresponding to the placement information and operated the robot. The robot successfully completed the entire series of passaging operations, from aspirating the medium from the old plate, washing with PBS, detaching the cells with trypsin, neutralizing and suspending the cells with medium, to seeding the cell suspension into the new plate. Microscopic observation of the new 6-well plate after the experiment confirmed that the cells were properly seeded, had attached normally, and were proliferating, as shown in Fig. 5E. This result demonstrates that Labware-Layout Planner possesses high versatility, capable of automatically generating effective initial layouts not only for simple dispensing tasks but also for complex, multi-step biological experiments like cell culture on different robot platforms.

Discussion

In this study, we developed Labware-Layout Planner, a system that automatically generates initial labware layouts and executable code from experimental protocols described in natural language. This represents a novel approach to alleviating the burden of a preparatory task in laboratory automation that has often been overlooked: the initial placement of labware, a task that has traditionally been left to the experience and tacit knowledge of researchers. This system automates a series of processes, including labware extraction using an LLM (labware-extractor) and constraint-based placement determination (position-planner and constraint-validator). This enables experimenters to start experiments with only natural language instructions and reagent preparation, without having to worry about complex placement problems, thereby further advancing laboratory automation. In this paper, we validated the effectiveness of this system under three different conditions: a simple colored water mixing experiment using the Opentrons OT-2, a complex qPCR reagent preparation experiment using the Opentrons OT-2, and a cell passaging experiment using the general-purpose humanoid robot, Maholo LabDroid.

In validations using the Opentrons OT-2, an automated liquid handler, our system demonstrated its effectiveness in tasks ranging from a simple colored water mixing experiment to the creation of a qPCR mix, which requires an understanding of molecular biology. First, in the colored water mixing experiment, the system was able to correctly dispense colored water into a 96-well plate based on the labware position information generated by Labware-Layout Planner and the motion script generated using an LLM. Furthermore, even in a more practical qPCR reagent preparation experiment, which involves a large variety and number of reagents and requires frequent pipette tip changes, the experiment was completed as intended while suppressing contamination and non-specific amplification. Moreover, its effectiveness was also validated through a cell passaging experiment using the dual-arm robot Maholo.[15] Maholo is a general-purpose humanoid robot with over 180 available slots for placing labware. When the robot was operated based on the placement information output by Labware-Layout Planner, it successfully completed the entire series of cell passaging operations. Microscopic observation after the experiment confirmed that the cells were properly seeded, attached, and proliferating. These results indicate that our system is not limited to a specific robot platform or task but is also adaptable to biological experiments using more flexible dual-arm robots. This suggests the potential to significantly reduce the burden of diverse initial experimental setup tasks, thereby contributing to an environment where researchers can focus more on their core research activities.

This study is positioned within the recent trend of applying large language models (LLMs) to autonomous scientific research tasks.[14,17,18] It is known that by combining goal-oriented instructions with an error-correction loop, LLMs can

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

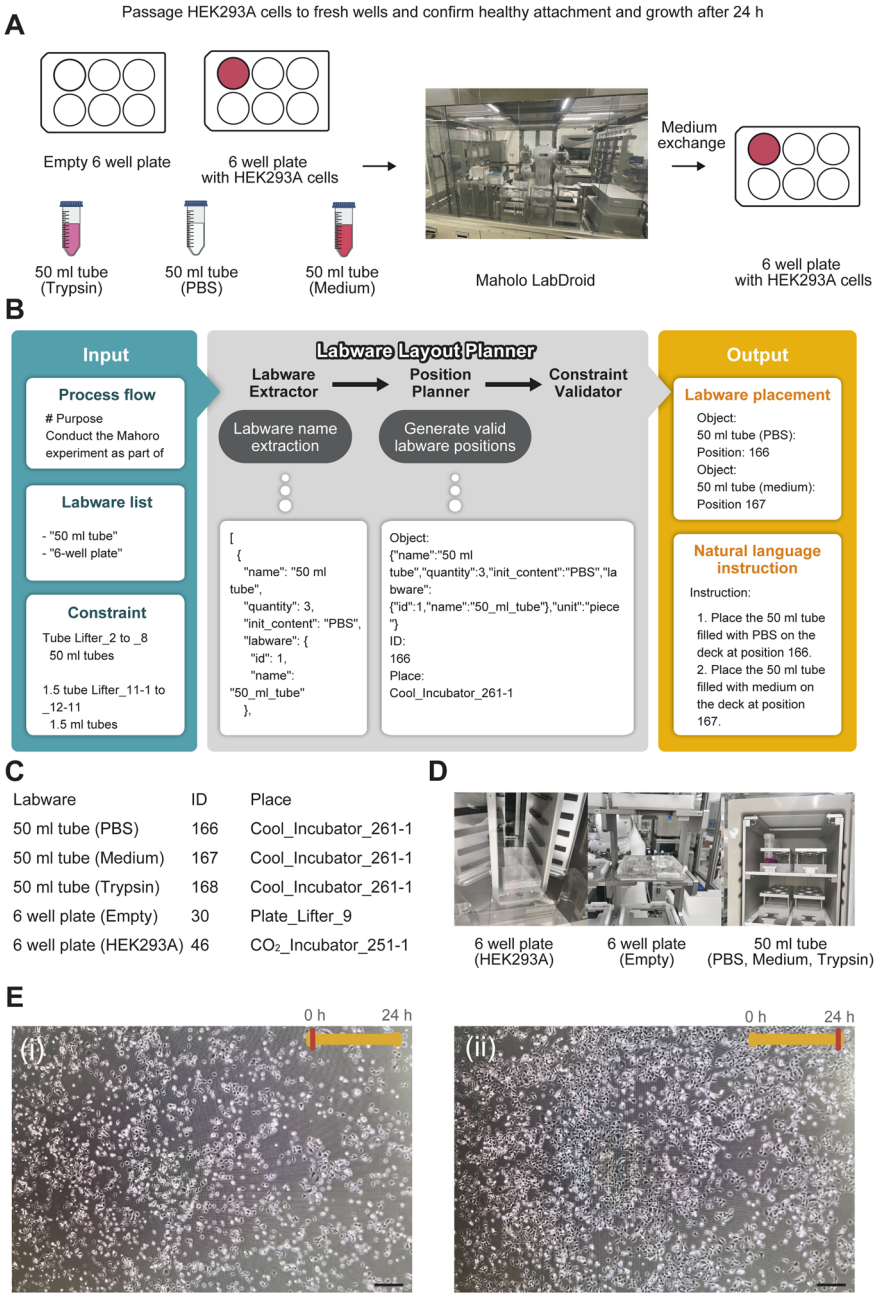

Fig. 5 Validation of Labware-Layout Planner in a cell passaging experiment using the Maholo robot. (A) Schematic of the cell culture experimental protocol. HEK293A cells are passaged from a 6-well plate to a new one after media exchange, executed by the Maholo LabDroid (dual-armed robot). (B) Detailed workflow using Labware-Layout Planner. Inputs (natural language process flow, available labware list, constraints for Maholo) are processed to generate outputs (a labware placement map and natural-language setup instructions). Examples of actual intermediate outputs (JSON) are shown. (C) The labware placement map and natural-language setup instructions generated by Labware-Layout Planner for the passaging workflow. (D) Manual implementation performed according to (C): a photograph and corresponding schematic of the labware setup on the Maholo experimental platform. (E) Microscope images of the experimental outcome. (i) Shows a well immediately after seeding cells, and (ii) image shows a well 24 hours after. Scale bars indicate 300 μm.

generate executable Python scripts for the OT-2 with high accuracy.[13] Additionally, attempts have been made to use LLMs to extract biological experiment protocols from literature in a unified format and make them executable on a thermal cycler.[17] Whereas these prior studies primarily focused on the logical interpretation of natural language protocols and their conversion into executable code, our research is unique in that it addresses the often-overlooked problem of the physical initial placement of labware. Specifically, we present a framework that automatically generates a layout considering real-world feasibility and safety, such as preventing cross-contamination and optimizing robotic arm movement efficiency, by incorporating

© 2026 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

physical constraints like absolute and relative positions, a task previously determined manually by humans. In this respect, our work goes beyond mere code generation to include the physical setup of the experiment within the scope of automation. It thus complements existing approaches and makes a significant contribution toward realizing more practical, real-world laboratory automation. In this context, our approach complements traditional symbolic planning frameworks, such as Task and Motion Planning (TAMP).[19] While TAMP excels at generating valid sequences of actions given a structured state, it cannot inherently process unstructured natural language inputs. Recent work has begun to explore using LLMs to translate natural language into intermediate representations that TAMP algorithms can consume.[20,21] By utilizing an LLM for initial semantic parsing, our system bridges this gap in the laboratory domain, converting natural language protocols into the structured 'initial state' (labware layout) required for effective downstream motion planning. Furthermore, unlike manual design, it eliminates the cognitive overhead of combinatorial layout planning, ensuring reproducible and safe configurations without requiring human intervention.

However, several challenges that must be overcome for practical application also became apparent. First is the accuracy of the LLM-based item extraction (labware-extractor), which serves as the system's entry point. While it succeeded in 9 out of 10 trials for the simple colored water experiment, the success rate dropped to 50% (5 out of 10) for the qPCR reagent preparation, which involves a significantly larger variety and number of labware items. This indicates that as protocol complexity increases, the difficulty of the task for the LLM to accurately extract all items also increases, making the implementation of a validation mechanism to ensure the reliability of the extraction results essential. In this context, we also performed a controlled benchmark comparing LLP with off-the-shelf LLM prompting (SI Appendix "benchmark against off-the-shelf LLM prompting and repeatability"), showing higher constraint satisfaction and more consistent outputs under repeated identical inputs. Second is the complexity of setting constraints in the placement determination stage (position-planner and constraint-validator). A strength of our system is its ability to flexibly specify rules that reflect on-site practices, such as separating the trash bin from reagents. However, as constraints become more complex, the computational cost increases,[6] and the user is burdened with the additional effort of specifying these constraints. Furthermore, the act of enumerating relative constraints itself, such as listing all pairs of labware items that should not be adjacent, is currently a manual task and represents a substantial share of the effort required to deploy the system. At present, users must express these rules explicitly in the JSON constraint file. In future work we plan to add higher-level constraint templates and reusable constraint blocks that can be instantiated for new protocols, so that common patterns (for example, "separate all waste containers from reagents") can be specified once rather than at the level of individual item pairs. In addition, although position-planner already converts negative ("cannot be placed in") constraints into the corresponding allowed locations, the present interface only exposes this negative form to users; extending the interface to also accept positive ("must be placed in") constraints is a straightforward refinement that should further simplify constraint specification. Currently, adjacency is determined by assuming a grid-based layout, but a system based on coordinates and distances would be more practical. In the future, extensions such as the automatic generation and prioritization of constraints reflecting biological context, as well as dynamic adjustment of constraints, are desired to address these issues. Third, due to implementation constraints, our system requires placement candidates to be specified by integer values, simplifying the workspace into a 2D grid. However, real-world laboratory environments are inherently three-dimensional. Future extensions must ensure three-dimensional consistency, such as accounting for stacking height limits, vertical arm clearance, and *Z*-axis collision avoidance by incorporating 3D model information for all labware.

In addition to these individual challenges, a broader perspective on system design is also necessary. The Labware-Layout Planner developed in this study focuses on generating initial labware placement information. Clearly separating the labware layout determination process from the code generation process for individual robots is crucial for ensuring the system's versatility and reproducibility. This allows the generated placement information to be in a neutral and reusable format, independent of specific platforms like the OT-2 or Maholo. However, this decoupled architecture inevitably creates a gap between the generated placement information and the final executable code. In this study, we incidentally generated an OT-2 script to bridge this gap, but this conversion process itself could become a new bottleneck in lab automation. Indeed, in the qPCR experiment, when we tasked an LLM with generating OT-2 code based on a valid layout proposed by position-planner, we observed an inconsistency where the script was generated for a different labware layout. This phenomenon suggests a risk that the internal state consistency of the LLM may be compromised as protocols become more complex. In our experiments, this error-feedback loop only modestly improved the overall success rate for the complex qPCR protocol: even after multiple regeneration attempts, many scripts still contained syntactic or semantic errors. This suggests that, with the model available at the time of our study, the main limitation lay in the LLM's ability to produce fully consistent OT-2 code from long prompts, rather than in the absence of a feedback mechanism. We therefore expect that more recent LLMs, or a dedicated compiler-like module that maps the neutral placement representation to robot-specific instructions, will further reduce these errors and make the end-to-end pipeline more reliable. Therefore, the next critical step toward achieving end-to-end experimental automation will be the development of a compiler-like module that takes the neutral placement information from Labware-Layout Planner as input and reliably auto-generates executable code for a variety of robot platforms.

This approach has the potential to reduce the hidden human burden of labware placement, which has previously depended on the tacit knowledge of researchers. Our framework addresses this challenge by explicitly encoding and enforcing physical

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

constraints *via* a deterministic planner, enabling verifiable and robot-agnostic layout generation beyond prompt-based approaches. However, this study also highlighted challenges, including the accuracy of LLM-based item extraction, the gap between the generated placement information and the robot's executable code, and the issue of how to map real-world 3D placement candidates into the software. As future work, we believe the community would benefit from a standardized, versioned benchmark suite for life-science laboratory automation (fixed protocol sets, inventories/constraints, and validity/constraint-satisfaction metrics) to enable longitudinal tracking of progress, and we hope to contribute to such an effort. Furthermore, a controlled study comparing time-to-valid-layout and error rates with manual planning is an important next step. In the future, overcoming these challenges will require not only improving the reliability of the components but also developing modules compatible with diverse robot platforms. Ultimately, this approach is expected to contribute to the realization of the future laboratory, where researchers can seamlessly transition from describing an idea in natural language to its physical execution.

Author contributions

Conceptualization: H. O.; G. N. K.; K. O. Data curation: Y. T.-A.; T. I.; A. K.; K. A. Funding acquisition: H. O.; G. N. K.; K. T. Investigation: H. O.; Y. T.-A.; T. I.; A. K.; K. A. Methodology: Y. T.-A.; T. I. Project administration: H. O.; G. N. K.; A. K. Resources: H. O.; G. N. K.; K. O.; A. K.; K. A.; K. T. Software: Y. T.-A.; T. I. Supervision: H. O.; G. N. K.; A. K. Visualization: Y. T.-A.; T. I. Writing – original draft: Y. T.-A.; T. I. Writing – review & editing: H. O.; G. N. K.; K. O.; Y. T.-A.; T. I.; A. K.; K. A. Y. T.-A. and T. I. contributed equally and have the right to list their name first in their CV.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The source code for Labware-Layout Planner and all data required to reproduce the results and figures (Fig. 3–5 and Table 1) in this manuscript are publicly available at https://github.com/labauto/Labware-Layout-Planner and have been archived on Zenodo (https://doi.org/10.5281/zenodo.17254729).

Supplementary information (SI) is available. See DOI: https://doi.org/10.1039/d6dd00026f.

Acknowledgements

We thank Hiroko Uchida for creating the illustrations for the figures. This study was supported by the Medical Research Center Initiative for High Depth Omics, Nanken-Kyoten, and Multilayered Stress Diseases, Science Tokyo (JPMXP1323015483 to G. N. K.), the JST-Mirai Program (JPMJMI18G4 and JPMJMI20G7 to K. T. and H. O.), Grant-in-Aid for Early-Career Scientists (JP22K17992 to H. O.), Grant-in-Aid for JSPS Fellows (JP22KJ3148 to A. K.), Grant-in-Aid for Transformative Research Areas (A) (JP23H04149 to H. O.), Grant-in-Aid for Scientific Research (C) (JP23K11820 to G. N. K.), JST BOOST (JPMJBS2414 to Y. T.), JST K Program (JPMJKP25V1 to G. N. K.), support from the RIKEN TRIP Advanced General Intelligence for Science Program (AGIS) to K. T. and H. O., and JST CREST (JPMJCR2551 to G. N. K. and H. O.).

References

1 R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Sparkes, K. E. Whelan and A. Clare, *Science*, 2009, **324**, 85–89.
2 A. Angelopoulos, J. F. Cahoon and R. Alterovitz, *Sci. Robot.*, 2024, **9**, eadm6991.
3 K. Ochiai, Y. Tahara-Arai, A. Kato, K. Kaizu, H. Kariyazaki, M. Umeno, K. Takahashi, G. N. Kanda and H. Ozaki, *Digital Discovery*, 2025, **4**, 2285–2297.
4 M. A. Torres-Acosta, G. J. Lye and D. Dikicioglu, *Biochem. Eng. J.*, 2022, **188**, 108713.
5 J. N. Socea, V. N. Stone, X. Qian, P. L. Gibbs and K. J. Levinson, *Front. Public Health*, 2023, **11**, 1195581.
6 M. Ferdosi, Y. Ge and C. Kingsford, *Leibniz International Proceedings in Informatics (LIPIcs)*, 2023, vol. 273, DOI: 10.4230/LIPIcs.WABI.2023.23.
7 G. Wu, R. Wang and C. W. Coley, *Digital Discovery*, 2025, **4**(9), 2593–2601.
8 E. J. Chory, D. W. Gretton, E. A. DeBenedictis and K. M. Esvelt, *Mol. Syst. Biol.*, 2021, **17**, e9942.
9 Z. P. Harmer and M. N. McClean, *ACS Synth. Biol.*, 2023, **12**, 1943–1951.
10 G. Moukarzel, Y. Wang, W. Xin, C. Hofmann, A. Joshi, J. W. Loughney and A. Bowman, *SLAS Technol.*, 2024, **29**, 100205.
11 G. N. Kanda, T. Tsuzuki, M. Terada, N. Sakai and N. Motozawa, *Elife*, 2022, **11**, e77007.
12 I. Holland and J. A. Davies, *Front. Bioeng. Biotechnol.*, 2020, **8**, 571777.
13 T. Inagaki, A. Kato, K. Takahashi, H. Ozaki and G. N. Kanda, *arXiv*, 2023, preprint, arXiv:2304.10267, DOI: 10.48550/arXiv.2304.10267.
14 D. A. Boiko, R. MacKnight and G. Gomes, *arXiv*, 2023, preprint, arXiv:2304.05332, DOI: 10.48550/arXiv.2304.05332.
15 N. Yachie, Robotic Biology Consortium and T. Natsume, *Nat. Biotechnol.*, 2017, **35**, 310–312.
16 K. Kishimoto, K. T. Furukawa, A. Luz-Madrigal, A. Yamaoka, C. Matsuoka, M. Habu, C. Alev, A. M. Zorn and M. Morimoto, *Nat. Commun.*, 2020, **11**, 4159.
17 S. Jiang, D. Evans-Yamamoto, D. Bersenev, S. K. Palaniappan and A. Yachie-Kinoshita, *SLAS Technol.*, 2024, **29**, 100134.
18 A. M Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, *Nat. Mach. Intell.*, 2024, **6**, 525–535.
19 C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling and T. Lozano-Pérez, *Annu. Rev. Control Rob. Auton. Syst.*, 2021, **4**, 265–293.

© 2026 The Author(s). Published by the Royal Society of Chemistry

20 Y. Chen, J. Arkin, C. Dawson, Y. Zhang, N. Roy and C. Fan, in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, vol. 2, pp. 6695–6702.

21 K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid and N. Suenderhauf, 2023, preprint, arXiv:2307.06135, DOI: 10.48550/arXiv.2307.06135.

Open Access Article. Published on 14 May 2026. Downloaded on 6/13/2026 9:46:47 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.