BioMARS: A Multi-Agent Robotic System for Autonomous Biological Experiments

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) and vision-language models (VLMs) have the potential to transform biological research by enabling autonomous experimentation. Yet, their application remains constrained by rigid protocol design, limited adaptability to dynamic lab conditions, inadequate error handling, and high operational complexity. Here we introduce BioMARS (Biological Multi-Agent Robotic System), an intelligent platform that integrates LLMs, VLMs, and modular robotics to autonomously design, plan, and execute biological experiments. BioMARS uses a hierarchical architecture: the Biologist Agent synthesizes protocols via retrieval-augmented generation; the Technician Agent translates them into executable robotic pseudo-code; and the Inspector Agent ensures procedural integrity through multimodal perception and anomaly detection. The system autonomously conducts cell passaging and culture tasks, matching or exceeding manual performance in viability, consistency, and morphological integrity. It also supports context-aware optimization, outperforming conventional strategies in differentiating retinal pigment epithelial cells. A web interface enables real-time human-AI collaboration, while a modular backend allows scalable integration with laboratory hardware. These results highlight the feasibility of generalizable, AI-driven laboratory automation and the transformative role of language-based reasoning in biological research.

1 Introduction

The convergence of robotic automation and artificial intelligence is reshaping experimental biology, promising greater reproducibility, throughput, and independence from human variability Holland & Davies (2020). However, the complexity of biological protocols—which demand adaptive decision-making, multi-stage coordination, and interpretation of nuanced environmental feedback—has hindered the realization of fully autonomous systems. Existing automation solutions, ranging from specialized liquid handling robots Dettinger et al. (2022); Novak et al. (2020); Taguchi et al. (2023), to modular single-arm platforms for cell culture automation Hamm et al. (2024); Tristan et al. (2021), and dual-arm platforms enabling automated cell production Königer et al. (2024); Yachie & Natsume (2017); Ochiai et al. (2021), often require extensive manual oversight and lack the flexibility to navigate unanticipated procedural deviations. Early systems focused on streamlining specific tasks, including biofoundries Chao et al. (2017), IoT-enabled experimental platforms Miles & Lee (2018), and clinical sample preparation Müller et al. (2020), but these non-robotic arm systems still faced hardware limitations that prompted the development of robotic arm solutions.

Concurrently, large language models (LLMs) and vision–language models (VLMs) are transforming scientific problem-solving by enabling machines to parse literature, synthesize knowledge, and execute multi-modal reasoning across diverse domains Vaswani et al. (2017); Wang et al. (2024); Luu & Buehler (2024); Zhang et al. (2025). Recent efforts leveraging LLMs in chemical experimentation Boiko et al. (2023); Darvish et al. (2025); Cooper et al. (2025) and biological protocol generation O'Donoghue et al. (2023); Huang et al. (2024) signal a paradigm shift toward AI-native experimentation. Yet, their integration with physical robotic systems for biological execution remains underexplored.

Here we introduce BioMARS (Biological Multi-Agent Robotic System), a dual-arm robotic platform orchestrated by LLMs and VLMs Zhu et al. (2023); Zhang et al. (2024) for fully autonomous execution of biological experiments. BioMARS performs end-to-end protocol design, environmental coordination, and robotic manipulation through adaptive multimodal reasoning. By converting research literature into actionable procedures and coupling them with error-aware execution strategies, the system ensures both flexibility and robustness in complex biological tasks.

We demonstrate BioMARS across five experimental capabilities: (1) efficiently searching and analyzing online research documentation to design experimental protocols for diverse cell types under varying conditions; (2) accurately translating and executing these protocols using a dual-arm biological laboratory; (3) detecting experimental errors via keyframe analysis; (4) performing end-to-end cell culturing; and (5) resolving optimization issues through the analysis of historical experimental data.

2 Related Work

2.1 Automation in Life Science Research Laboratory

Automation in the life sciences has advanced from fixed-function devices toward increasingly integrated systems. Early non-robotic approaches, such as biofoundries for synthetic biology Chao et al. (2017), IoT-enabled platforms with closed-loop feedback Miles & Lee (2018), and automated clinical preparation systems Müller et al. (2020), standardized workflows but were limited by rigid hardware. Open-source pipetting robots Dettinger et al. (2022) introduced customization but lacked physical adaptability.

Subsequent progress in robotic-arm systems expanded precision and scalability. Single-arm systems advanced through modular and vision-assisted designs for dynamic pipetting and culture handling Li et al. (2023); Zhang et al. (2023); Hamm et al. (2024). Dual-arm systems further improved coordination and throughput, enabling continuous tissue fabrication and multi-line cell maintenance through adaptive scheduling Königer et al. (2024); Ochiai et al. (2021). These developments laid the groundwork for intelligent, general-purpose platforms but remain constrained by static programming and limited semantic understanding of experimental intent.

2.2 Large Model based Multi-agent Systems

The emergence of LLMs has revitalized multi-agent system research. Foundational work in agent communication and coordination Wooldridge & Jennings (1995) now converges with LLM-driven reasoning, enabling distributed collaboration across virtual and robotic agents. Systems such as Gorilla Patil et al. (2025) and RoCo Mandi et al. (2024) illustrate how LLMs can orchestrate multi-agent tasks through natural language, while domain-specific applications span materials discovery Zheng et al. (2023) and behavioral simulations Aher et al. (2023). These advances establish LLMs as both communication intermediaries and autonomous planners in multi-robot ecosystems.

2.3 LLMs and VLMs as Robot Planners

In robotic task planning, LLMs support semantic decomposition of goals Song et al. (2023); Silver et al. (2024) and direct code generation for control policies Liang et al. (2023); Huang et al. (2023); Wu et al. (2023). Integration with VLMs augments perceptual grounding and adaptability in dynamic settings, with systems such as ReplanVLM Mei et al. (2024) achieving real-time correction via visual feedback. Physically grounded multimodal frameworks Gao et al. (2024); Wake et al. (2024) demonstrate the feasibility of translating perception–action reasoning into robotic execution—an essential foundation for autonomous biological experimentation.

3 Architecture of BioMARS system

BioMARS (Biological Multi-Agent Robotic System) enables end-to-end autonomous execution of biological experiments through a network of specialized LLM- and VLM-based agents (Fig. 1a). Built on an enhanced Agentic Retrieval-Augmented Generation (RAG) framework with modular error correction Singh et al. (2025), BioMARS decomposes complex protocols, interprets unstructured literature, and dynamically synthesizes findings into executable procedures.

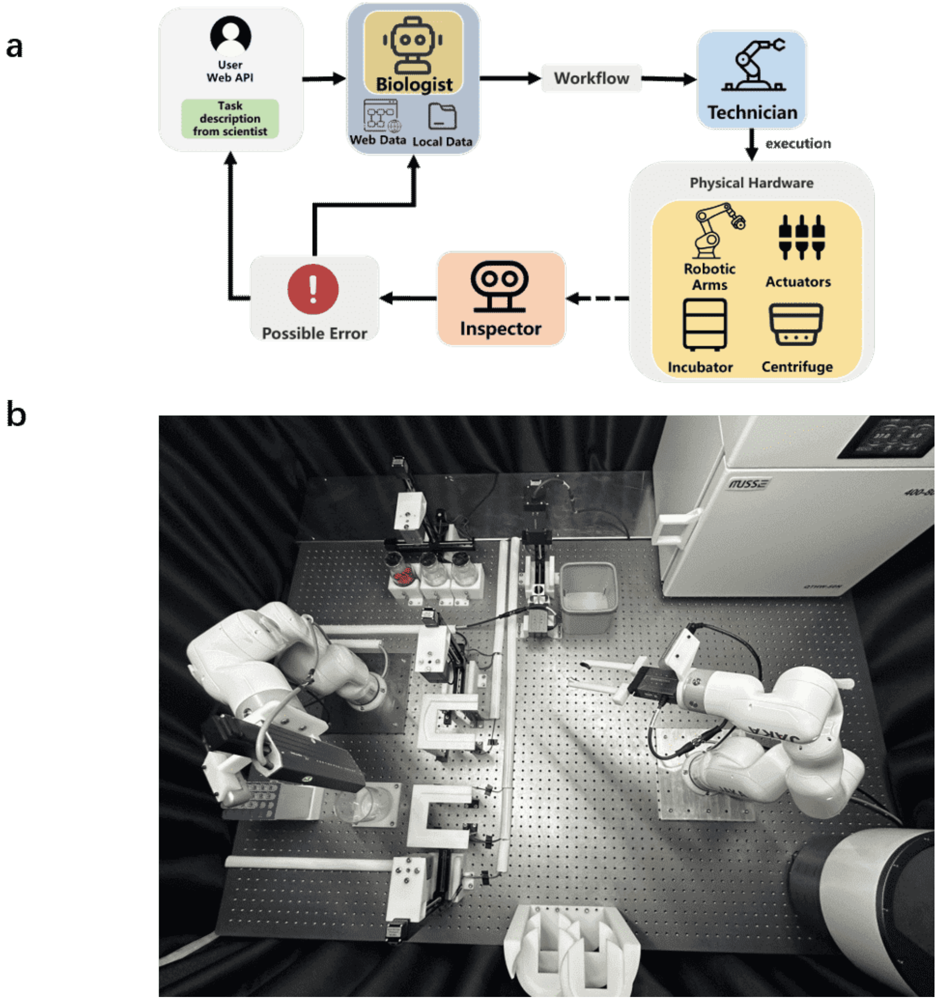


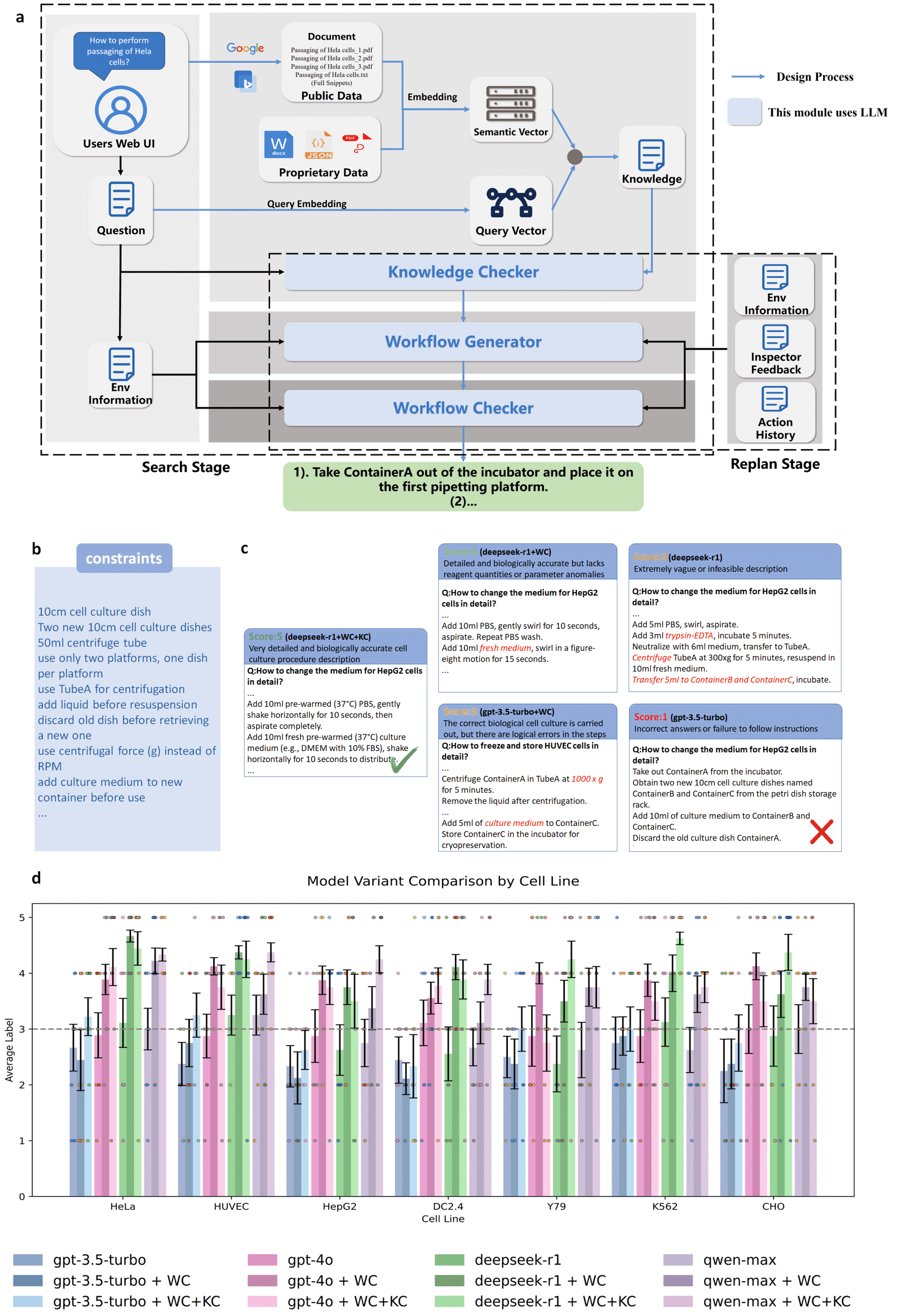
Figure 1: **System architecture and robotic setup. a,** Multi-agent workflow of BioMARS, comprising Biologist, Technician, and Inspector agents. **b,** Dual-arm robotic platform configured for autonomous biological experimentation.

The Biologist Agent ingests diverse open-access research documents, generates executable protocol steps by leveraging biological domain knowledge to create structured, constraint-aware queries. By incorporating constraints such as container type (e.g., petri dishes, flasks) and platform capacity, it tailors each protocol to the laboratory's operational environment. The Technician Agent transforms high-level plans into fine-grained control primitives for robotic execution. These primitives are allocated across dual robotic arms and coordinated with environmental modules such as incubator and centrifuge.

To ensure execution robustness, the Inspector Agent—powered by ViTs and VLMs—performs rapid anomaly detection. It identifies procedural deviations including geometric misalignments (e.g., unattached pipette tips, misaligned petri dishes) and mechanical failures, prompting replanning or user notification. This tri-agent system mirrors the modularity and task specialization seen in other autonomous platforms Boiko et al. (2023); M. Bran et al. (2024), enabling BioMARS to operate adaptively under changing experimental conditions.

The platform supports natural language prompts (e.g., "How to passage HeLa cells") via a web interface. Users can initiate, monitor, and modify experiments interactively. Critically, BioMARS's modular architecture allows seamless integration of new hardware and protocol domains through programmable function modules, facilitating extensibility across diverse biological workflows.

3.1 Protocol Synthesis under Environmental Constraints



10cm cell culture dish
Two new 10cm cell culture dishes
50ml centrifuge tube
use only two platforms, one dish per platform
use TubeA for centrifugation
add liquid before resuspension
discard old dish before retrieving a new one
use centrifugal force (g) instead of RPM
add culture medium to new container before use
...

Score:5 (deepseek-r1+WC+KC)
Very detailed and biologically accurate cell culture procedure description

Q:How to change the medium for HepG2 cells in detail?
...
Add 10ml pre-warmed (37°C) PBS, gently shake horizontally for 10 seconds, then aspirate completely.
Add 10ml fresh pre-warmed (37°C) culture medium (e.g., DMEM with 10% FBS), shake horizontally for 10 seconds to distribute. ✓
...

Score:4 (deepseek-r1+WC)
Detailed and biologically accurate but lacks reagent quantities or parameter anomalies

Q:How to change the medium for HepG2 cells in detail?
...
Add 10ml PBS, gently swirl for 10 seconds, aspirate. Repeat PBS wash.
Add 10ml *fresh medium*, swirl in a figure-eight motion for 15 seconds.
...

Score:3 (gpt-3.5-turbo+WC)
The correct biological cell culture is carried out, but there are logical errors in the steps

Q:How to freeze and store HUVEC cells in detail?
...
Centrifuge ContainerA in TubeA at *1000 x g* for 5 minutes.
Remove the liquid after centrifugation.
...
Add 5ml of *culture medium* to ContainerC.
Store ContainerC in the incubator for cryopreservation.

Score:2 (deepseek-r1)
Extremely vague or infeasible description

Q:How to change the medium for HepG2 cells in detail?
...
Add 5ml PBS, swirl, aspirate.
Add 3ml *trypsin-EDTA*, incubate 5 minutes.
Neutralize with 6ml medium, transfer to TubeA.
Centrifuge TubeA at 300xg for 5 minutes, resuspend in 10ml fresh medium.
Transfer 5ml to ContainerB and ContainerC, incubate.

Score:1 (gpt-3.5-turbo)
Incorrect answers or failure to follow instructions

Q:How to change the medium for HepG2 cells in detail?
Take out ContainerA from the incubator.
Obtain two new 10cm cell culture dishes named ContainerB and ContainerC from the petri dish storage rack.
Add 10ml of culture medium to ContainerB and ContainerC.
Discard the old culture dish ContainerA. ✗

Figure 2: **Biologist Agent architecture and evaluation. a,** Biologist agent pipeline integrating document retrieval, semantic matching, and workflow refinement under constraints. **b,** Representative experimental constraints. **c,** Example protocol outputs with scores and errors. **d,** Performance comparison of four models (GPT-3.5-Turbo, GPT-4o, Deepseek-R1, Qwen-Max) and their variants on seven cell lines.

Reliable generation of biological protocols from literature poses challenges due to procedural complexity, heterogeneous experimental conditions, instrumentation constraints, and output formatting requirements. BioMARS addresses this through a multi-agent reasoning framework that integrates LLM-based planning with vector-based retrieval and verification mechanisms to generate biologically accurate, context-aware procedures.

At the center of this system is the Biologist Agent, which operates within an enhanced Agentic Retrieval-Augmented Generation (RAG) architecture (Fig. 2a). The agent retrieves relevant knowledge using online query APIs (Google and Bing), extracting three PDFs and three high-relevance web snippets per query. Full paragraphs associated with each snippet are selected to preserve semantic context. These passages, along with the embedded user query (using OpenAI's Ada model), undergo vector similarity ranking. The top five text chunks are used as context for downstream protocol generation.

Protocol construction is distributed across three sub-agents: the Knowledge Checker (KC), which filters domain-inconsistent content; the Workflow Generator (WG), which formulates stepwise procedures; and the Workflow Checker (WC), which iteratively refines outputs for logical coherence. The system accounts for laboratory constraints, such as limited stock of specific containers (e.g. 10 cm culture dishes), pipette tip volume (10 ml), and robotic station limits, ensuring that all outputs are executable on the BioMARS platform.

System performance was evaluated using a 70-query benchmark comprising 10 procedural categories across seven cell lines, ranging from routine tasks (e.g., cell passaging, thawing) to complex protocols (e.g., 3D culture, apoptosis analysis). Following Boiko et al. Boiko et al. (2023), model outputs were scored on a 5-point scale: 5 for fully detailed and accurate procedures; 4 for biologically sound steps with minor omissions; 3 for logically flawed but conceptually plausible outputs; 2 for vague or infeasible workflows; and 1 for incorrect or non-compliant procedures. Outputs below a score of 3 were considered task failures. Fig. 2c presents representative outputs with annotations.

Without WC or KC modules, base models—including GPT-4o, Qwen-Max, and DeepSeek-R1—did not exceed a mean score of 3. GPT-3.5 Turbo consistently underperformed; in one instance, it misinterpreted "How to change the HepG2 culture medium" by suggesting disposal of viable dishes and initiating culture from scratch (score: 1). DeepSeek-R1 proposed cell redistribution via trypsinization (score: 2), demonstrating procedural confusion.

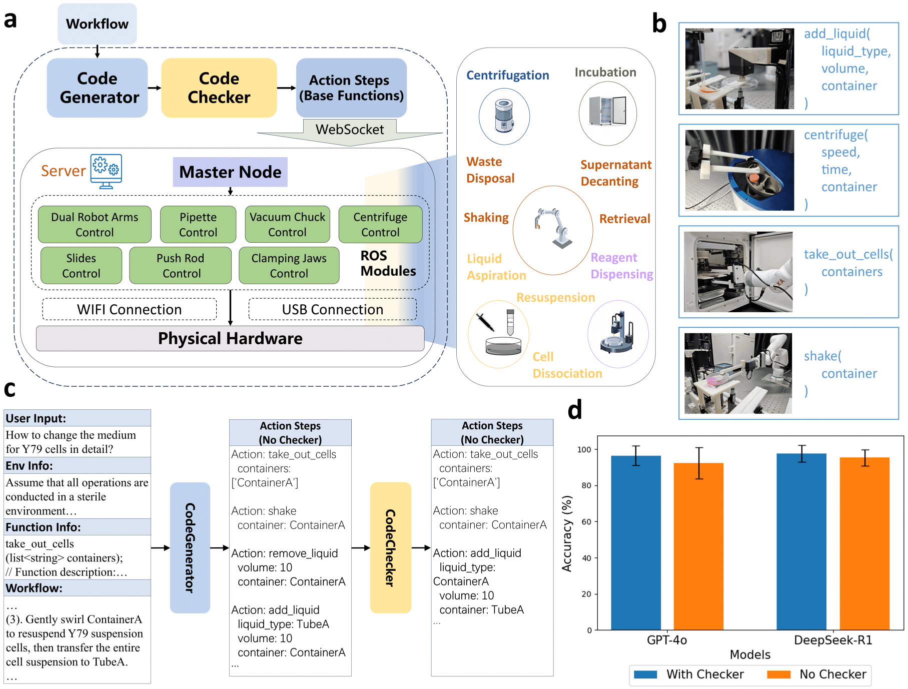
Incorporation of the WC module significantly improved structural logic. For example, DeepSeek-R1+WC successfully outlined PBS rinsing and medium replacement steps but omitted critical conditions (temperature, CO_2 levels), yielding a score of 4. Further integration with the KC module provided domain-specific validations: in the cryopreservation task for HUVECs, KC-corrected protocols mitigated centrifugation errors and ensured cryostorage in liquid nitrogen.

The best-performing configuration—DeepSeek-R1+WC+KC—achieved consistent scores of 5. Its output for HepG2 medium replacement detailed exact reagent volumes, environmental settings (37°C, 5% CO_2), and handling protocols (PBS rinse with horizontal agitation), aligning closely with expert protocols. These results affirm the critical role of domain validation (KC) and procedural refinement (WC) in transforming LLM outputs into executable, high-fidelity biological protocols (Fig. 2c,d).

3.2 Protocol-to-Code Translation for Robotics

Translating free-text experimental protocols into executable robotic commands remains a central bottleneck in laboratory automation. Existing systems typically rely on rigid, manually curated command sequences Kanda et al. (2022); Königer et al. (2024), which limits their adaptability to diverse and unstructured inputs. To address this constraint, we developed the Technician Agent—a dual-module system that autonomously interprets natural language protocols and converts them into validated robotic instructions.

The Technician Agent operates through a cooperative pipeline comprising a CodeGenerator and a CodeChecker module (Fig. 3a). The CodeGenerator, powered by an LLM, maps protocol descriptions into pseudo-code composed of primitive robotic operations such as `add_liquid`, `centrifuge`, and `shake` (Fig. 3b). The CodeChecker subsequently performs rule-based validation, enforcing functional correctness and environmental compatibility based on the predefined specification set.



```
add_liquid(
    liquid_type,
    volume,
    container
)

centrifuge(
    speed,
    time,
    container
)

take_out_cells(
    containers
)

shake(
    container
)
```

User Input:

How to change the medium for Y79 cells in detail?

Env Info:

Assume that all operations are conducted in a sterile environment…

Function Info:

take_out_cells (list<string> containers);
// Function description:…

Workflow:

…
(3). Gently swirl ContainerA to resuspend Y79 suspension cells, then transfer the entire cell suspension to TubeA.
…

Action Steps (No Checker)

```
Action: take_out_cells
  containers:
['ContainerA']

Action: shake
  container: ContainerA

Action: remove_liquid
  volume: 10
  container: ContainerA

Action: add_liquid
  liquid_type: TubeA
  volume: 10
  container: ContainerA
…
```

Action Steps (No Checker)

```
Action: take_out_cells
  containers:
['ContainerA']

Action: shake
  container: ContainerA

Action: add_liquid
  liquid_type:
ContainerA
  volume: 10
  container: TubeA
…
```

Figure 3: **Technician Agent architecture and performance of protocol translation and execution framework.** **a,** System workflow of Technician Agent, including CodeGenerator, CodeChecker, ROS node and the corresponding hardware module. **b,** Example pseudo-code instructions and corresponding robotic actions. **c,** The specific workflow of Technician Agent. **d,** Instruction accuracy comparison with and without CodeChecker for GPT-4o and DeepSeek-R1.

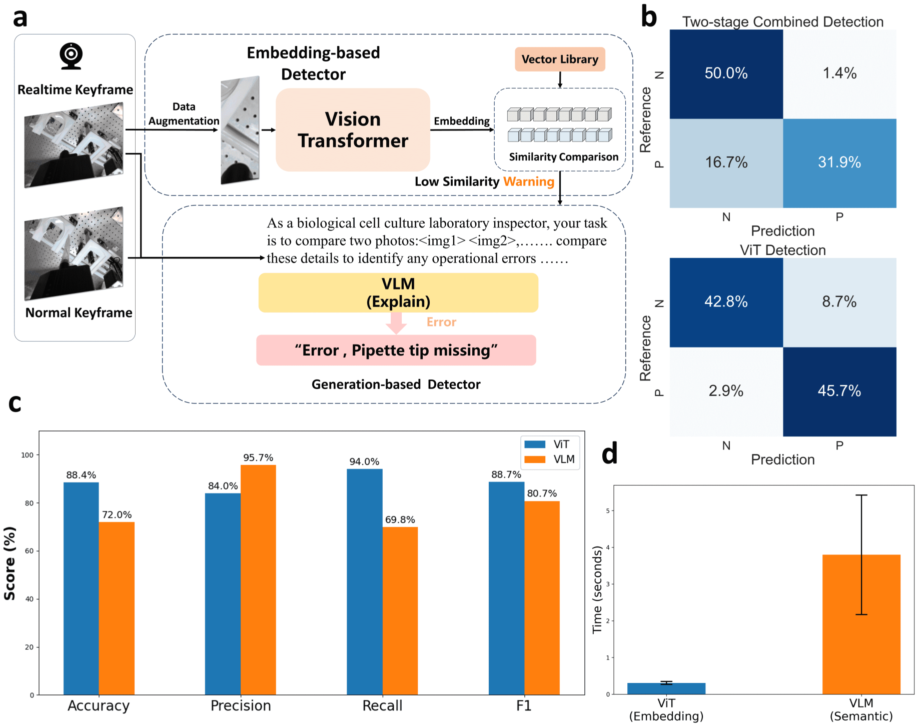
The pipeline structure is illustrated in Fig. 3c. Given a protocol input, the CodeGenerator produces candidate instructions tailored to the lab environment. These instructions are then parsed by the CodeChecker, which applies logical and semantic checks including parameter validation, function relevance, and argument structure. This ensures that all generated commands adhere to the operational and safety constraints of the BioMARS platform.

To assess performance, we benchmarked the Technician Agent across 300 experimental protocol steps. As shown in Fig. 3d, the full pipeline (CodeGenerator + CodeChecker, GPT-4o) achieved a 96.4% instruction-matching accuracy, outperforming a single-module baseline (92.4%). The impact of the CodeChecker module is particularly evident in complex procedural constructs. For example, when parsing the instruction “resuspend the cell pellet in 10 mL fresh complete growth medium,” the baseline failed to recognize the prerequisite transfer step. In contrast, the Technician Agent inserted an implicit `add_liquid` operation before resuspension, preserving procedural logic.

Beyond resolving implicit steps, the CodeChecker module also corrects parameter mismatches, enforces range constraints, and eliminates superfluous instructions. For instance, it detects and corrects overfilled volumes relative to container capacity and replaces invalid data types in function arguments. This systematic refinement substantially improves the robustness of the robotic instruction set.

By converting ambiguous natural language into explicit, verifiable pseudo-code, the Technician Agent enhances experimental reproducibility, reduces human error, and simplifies execution on robotic platforms. This capability shifts the experimental burden away from manual coding, enabling researchers to focus on scientific inquiry rather than operational encoding.

3.3 Hierarchical VLM-Based Error Detection



As a biological cell culture laboratory inspector, your task is to compare two photos:<img1> <img2>,……. compare these details to identify any operational errors ……

Figure 4: **Inspector Agent Overview and Performance Metrics. a,** Workflow diagram of the Technician Agent. **b,** Confusion matrix of two-stage combined detection and ViT detection. **c,** Performance of ViT and VLM on four evaluation metrics. **d,** The time performance of the two detection methods, ViT and VLM.

Biological experimentation demands strict precision, where minor procedural errors can compromise outcomes. Conventional automation platforms typically rely on basic object detection without semantic context awareness, limiting their robustness in dynamic laboratory environments Jiang et al. (2022). To address this, we developed the Inspector Agent—a hierarchical visual monitoring system integrating vision–language models (VLMs) and vision transformers (ViTs) Han et al. (2022) for multi-stage perception and error detection (Fig. 4a).

The first stage performs visual segmentation of experimental scenes using few-shot prompting with a VLM. Key objects—such as pipette tips, culture plates, and tubes—are segmented from raw RGB inputs. To enhance spatial resolution and minimize background interference, the bounding boxes generated by the VLM are manually refined. These cropped subregions are converted to grayscale, preserving structural cues like pipette orientation and tube angles while reducing color-based noise.

In the second stage, a ViT-based keyframe detection module encodes 23 visually discriminative actions (selected from 11 control primitives) into a reference embedding library. This module enables sub-second recognition of procedural steps. In benchmark tests, ViT achieved a mean inference latency of 0.3066 s - 91. 9% faster than GPT-4o (3.7960 s) - with lower temporal variability (coefficient of variation: 13.08% vs. 42.80%; Fig. 4d). In real-world experimental settings, the ViT achieved an F1 score of 88.7% and a recall of 94.0%, demonstrating high temporal stability and operational fidelity (Fig. 4c).

A final stage introduces zero-shot semantic validation using the VLM. When the ViT flags anomalies, frames are semantically compared with idealized keyframes using language-guided prompts (e.g., “attach pipette tip”). This semantic differential analysis enables detection of contextual errors beyond geometry alone. In validation, this mechanism achieved 95.7% precision and 80.7% F1 score(Fig. 4c), reducing the false positives rate from 8.7% to 1.4%—an 83% improvement (Fig. 4b). For example, detecting a detached pipette tip without a visible pipette is correctly flagged as an action violation. Upon confirmation, robotic operations are automatically paused and visual alerts issued.

By integrating geometric and semantic vision processing, the Inspector Agent ensures procedural robustness, accelerates feedback response times, and significantly reduces downstream execution failures.

4 Biological Experiment Design

4.1 Integrated Biological Experiment Design

To evaluate the biological reliability and operational efficiency of BioMARS, we conducted a comparative study between automated and manual cell passaging protocols across three representative cell types: HeLa (adherent), Y79 (suspension), and DC2.4 (semi-adherent/suspension). Experimental evaluation included metabolic viability, survival consistency, morphological preservation, and coefficient of variation (CV) analysis. All workflows adhered to established protocols, with BioMARS dynamically adapting process parameters to each cell line.

Cells were cultured in standard media: HeLa in DMEM with 10% FBS and 1% penicillin–streptomycin, Y79 in RPMI-1640 with 20% FBS and DC2.4 in RPMI-1640 with 10% FBS - under 5% CO2 at 37 ° C. Media changes were performed every 2–3 days. For passaging, adherent cells were detached with 0.25% trypsin–EDTA. The BioMARS system adjusted enzymatic digestion time and centrifugation based on cell type: 6 minutes for HeLa and 3 minutes for Y79, ensuring optimal yield and viability.

Metabolic viability was assessed 48 hours post-passaging using the CCK-8 assay. Optical density (OD) measurements showed no significant difference between BioMARS and manual protocols across all three cell types (Fig. 5d), indicating that automated processing maintained normal cellular proliferation. CV analysis revealed enhanced reproducibility in the BioMARS group: HeLa and Y79 samples exhibited 12–18% lower variability compared to manual handling (Fig. 5e).

Live/dead staining confirmed high post-passaging viability, with over 92% concordance between automated and manual groups (Fig. 5a,b). Green fluorescence indicated dominant live-cell populations, with clear contrast to the red-stained positive control. Morphological evaluation (Fig. 5c)

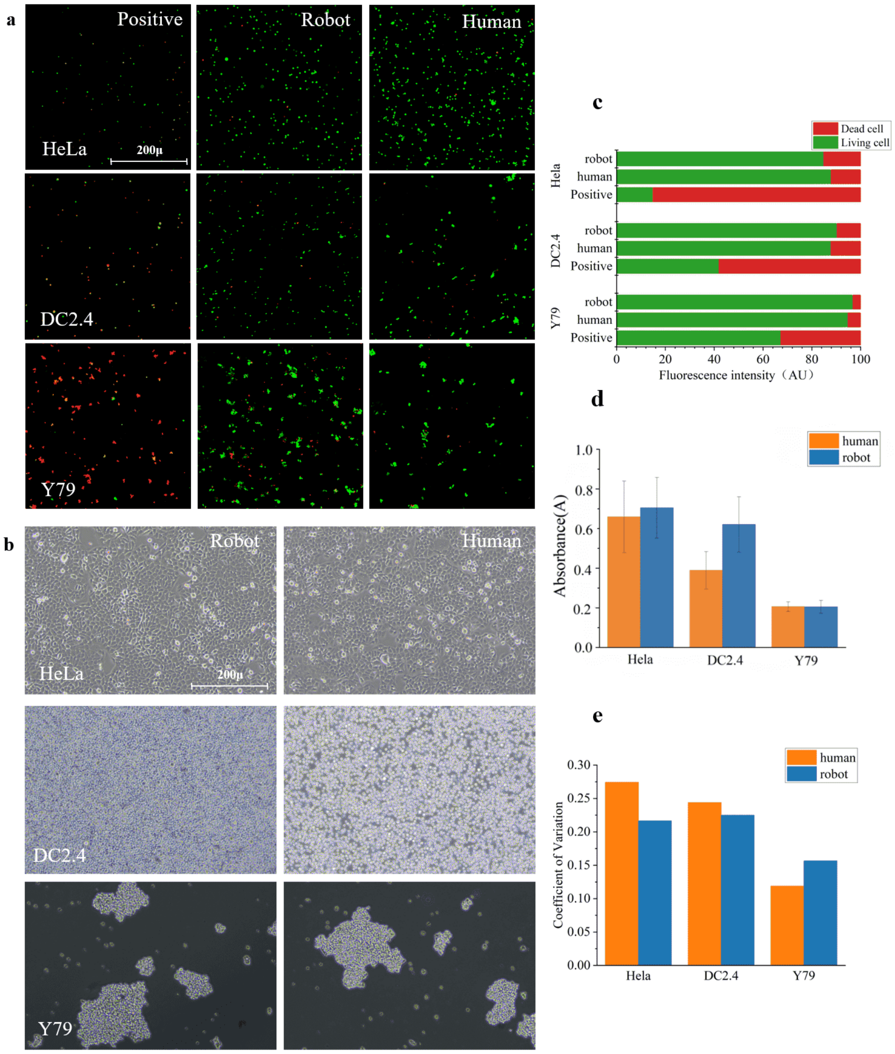
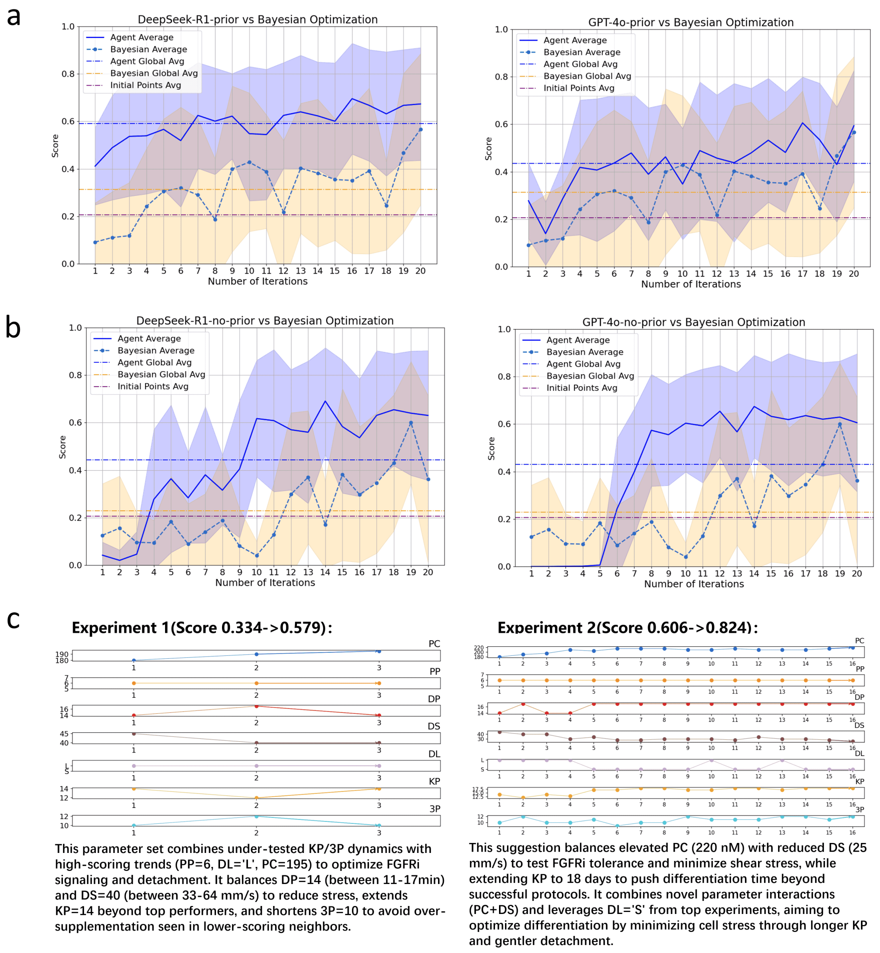


Figure 5: **Comparison of automated vs. manual cell passaging outcomes. a,** Fluorescence images of live/dead-stained cells (automated vs. manual) at 48 h post-passaging. **b,** Bright-field images of cell morphology post-passaging. **c,** Live/dead cell ratio comparison after passaging. **d,** Cell viability comparison between methods. **e,** CV of CCK-8 viability across repeats (reproducibility).

showed no detectable structural abnormalities, further confirming the BioMARS system's ability to preserve cell integrity.

In addition to biological fidelity, BioMARS markedly improved operational efficiency. Manual passaging required approximately 60 minutes per cell line, whereas the BioMARS system reduced hands-on time to 5–8 minutes—representing a 90% reduction. This time savings translates into higher throughput and improved standardization, minimizing human error and procedural variability.

Collectively, these results establish that BioMARS performs comparably or superior to manual protocols in biological outcome metrics while offering significant gains in consistency, reproducibility, and efficiency.



Experiment 1(Score 0.334->0.579):

This parameter set combines under-tested KP/3P dynamics with high-scoring trends (PP=6, DL='L', PC=195) to optimize FGFRi signaling and detachment. It balances DP=14 (between 11-17min) and DS=40 (between 33-64 mm/s) to reduce stress, extends KP=14 beyond top performers, and shortens 3P=10 to avoid over-supplementation seen in lower-scoring neighbors.

Experiment 2(Score 0.606->0.824):

This suggestion balances elevated PC (220 nM) with reduced DS (25 mm/s) to test FGFRi tolerance and minimize shear stress, while extending KP to 18 days to push differentiation time beyond successful protocols. It combines novel parameter interactions (PC+DS) and leverages DL='S' from top experiments, aiming to optimize differentiation by minimizing cell stress through longer KP and gentler detachment.

Figure 6: **Results of the iPSC-RPE optimization experiments.** **a,** Performance comparison between DeepSeek-R1 and GPT-4o models versus Bayesian Optimization using 10 prior experimental data points. **b,** Comparative analysis of DeepSeek-R1 and GPT-4o against Bayesian Optimization without leveraging prior experimental knowledge. **c,** Parameter recommendations from LLM-based optimizers across selected iteration rounds.

4.2 Biological Optimizing Capability

Beyond static protocol generation, the Biologist Agent was evaluated for its capacity to perform biological optimization—an advanced task requiring iterative reasoning, mechanistic understanding, and strategic parameter adjustment. We assessed this capability using a publicly available dataset for optimizing differentiation efficiency of induced pluripotent stem cell-derived retinal pigment epithelial (iPSC-RPE) cells Kanda et al. (2022), which defines a high-dimensional experimental space grounded in biological constraints.

The optimization target was the pigment score, a key phenotypic marker of iPSC-RPE maturation. Seven tunable parameters were considered across preconditioning, detachment, and differentiation stages: FGFRi concentration (PC: 0–505 nM) and exposure duration (PP: 1–6 days); trypsin incubation time (DP: 5–23 min), pipetting strength (DS: 10–100 mm / s) and pipetting length (DL: short/long); KSR withdrawal schedule (KP: 1–19 days); and three-supplement exposure duration (3P: 3–19 days). This setup presents a biologically grounded, combinatorially complex optimization challenge.

To simulate realistic experimental conditions, optimization was constrained to 20 iterations, initialized from 10 randomly selected low-performing conditions (pigment score ¡ 0.6). Parameter selection used KDTree-based nearest-neighbor interpolation Friedman et al. (1977), with outputs formatted in structured JSON for reproducibility. We compared three strategies: DeepSeek-R1, GPT-4o, and Bayesian optimization under identical initialization settings.

In the prior-informed setting (Fig. 6a), DeepSeek-R1 consistently outperformed baselines, reaching a final pigment score of 0.5913—surpassing GPT-4o (0.4344; +15.8%) and Bayesian optimization (0.3130; +28.5%). By iteration 7, it achieved 0.6252 and continued steady improvement. GPT-4o plateaued at 0.606, while Bayesian optimization peaked early at 0.5671. DeepSeek-R1's advantage stems from its ability to encode mechanistic constraints; for instance, in one high-scoring trial (Fig. 6c), it selected PC = 220 nM (balancing efficacy and toxicity), DS = 25 mm/s (minimizing shear stress), and KP = 18 days (prolonging Wnt signaling), reflecting domain-consistent reasoning.

GPT-4o occasionally produced viable configurations but lacked consistent convergence, likely due to reliance on pretrained heuristics. Bayesian optimization, devoid of biological priors, frequently proposed implausible combinations (e.g., PC = 405.17 nM; KP = 2 days), resulting in limited progress.

In the no-prior setting (Fig. 6b), DeepSeek-R1 again demonstrated robust generalization, reaching performance comparable to the prior-informed case. GPT-4o improved after iteration 8, ultimately reaching a moderate score of 0.6303. Bayesian optimization showed minimal learning, with scores remaining near baseline. DeepSeek-R1 also exhibited superior balance between exploration and exploitation, as evidenced by a lower standard deviation in output scores (0.2366 vs. 0.2447 for GPT-4o and 0.2785 for Bayesian optimization), enabling more stable convergence.

These results validate the potential of knowledge-integrated LLMs to optimize complex biological systems under data-sparse conditions. By combining contextual reasoning with structured decision-making, such agents reduce dependency on manual tuning and offer scalable solutions for experimental design. Future directions include reinforcement learning frameworks to further enhance adaptive feedback integration in regenerative biology workflows.

References

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pp. 337–371. PMLR, 2023.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Ran Chao, Shekhar Mishra, Tong Si, and Huimin Zhao. Engineering biological systems using automated biofoundries. *Metabolic Engineering*, 42:98–108, 2017.

Andrew Cooper, Zhengxue Zhou, Satheeshkumar Veeramani, Francisco Galeano, and Hatem Fakhruldeen. Lira: Localization, inspection, and reasoning module for autonomous workflows in self-driving labs. 2025.

Kourosh Darvish, Marta Skreta, Yuchi Zhao, Naruki Yoshikawa, Sagnik Som, Miroslav Bogdanovic, Yang Cao, Han Hao, Haoping Xu, Alán Aspuru-Guzik, et al. Organa: a robotic assistant for automated chemistry experimentation and characterization. *Matter*, 8(2), 2025.

Philip Dettinger, Tobias Kull, Geethika Arekatla, Nouraiz Ahmed, Yang Zhang, Florin Schneiter, Arne Wehling, Daniel Schirmacher, Shunsuke Kawamura, Dirk Loeffler, et al. Open-source personal pipetting robots with live-cell incubation and microscopy compatibility. *Nature Communications*, 13(1):2999, 2022.

Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3 (3):209–226, 1977.

Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. In *2024*

IEEE International Conference on Robotics and Automation (ICRA), pp. 12462–12469. IEEE, 2024.

Jungmin Hamm, Seonghyeon Lim, Jiae Park, Jiwon Kang, Injun Lee, Yoongeun Lee, Jiseok Kang, Youngjun Jo, Jaejin Lee, Seoyeong Lee, et al. A modular robotic platform for biological research: Cell culture automation and remote experimentation. *Advanced Intelligent Systems*, 6(5): 2300566, 2024.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

Ian Holland and Jamie A Davies. Automation in the life science research laboratory. *Frontiers in bioengineering and biotechnology*, 8:571777, 2020.

Kaixuan Huang, Yuanhao Qu, Henry Cousins, William A Johnson, Di Yin, Mihir Shah, Denny Zhou, Russ Altman, Mengdi Wang, and Le Cong. Crispr-gpt: An llm agent for automated design of gene-editing experiments. *arXiv preprint arXiv:2404.18021*, 2024.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.

Genki N Kanda, Taku Tsuzuki, Motoki Terada, Noriko Sakai, Naohiro Motozawa, Tomohiro Masuda, Mitsuhiro Nishida, Chihaya T Watanabe, Tatsuki Higashi, Shuhei A Horiguchi, et al. Robotic search for optimal cell culture in regenerative medicine. *Elife*, 11:e77007, 2022.

Lukas Königer, Christoph Malkmus, Dalia Mahdy, Thomas Däullary, Susanna Götz, Thomas Schwarz, Marius Gensler, Niklas Pallmann, Danjouma Cheufou, Andreas Rosenwald, et al. Rebia—robotic enabled biological automation: 3d epithelial tissue production. *Advanced Science*, 11(45):2406608, 2024.

Kai Li, WenHui Huang, HaiTao Guo, YanYan Liu, Shuxian Chen, Heng Liu, and Qi Gu. Advancements in robotic arm-based 3d bioprinting for biomedical applications. *Life Medicine*, 2(6): lnad046, 2023.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Rachel K Luu and Markus J Buehler. Bioinspiredllm: Conversational large language model for the mechanics of biological and bio-inspired materials. *Advanced Science*, 11(10):2306724, 2024.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–299, 2024. doi: 10.1109/ICRA57147.2024.10610855.

Aoran Mei, Guo-Niu Zhu, Huaxiang Zhang, and Zhongxue Gan. Replanvlm: Replanning robotic tasks with visual language models. *IEEE Robotics and Automation Letters*, 2024.

Ben Miles and Peter L Lee. Achieving reproducibility and closed-loop automation in biological experimentation with an iot-enabled lab of the future. *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, 23(5):432–439, 2018.

Torsten Müller, Mathias Kalxdorf, Rémi Longuespée, Daniel N Kazdal, Albrecht Stenzinger, and Jeroen Krijgsveld. Automated sample preparation with sp 3 for low-input clinical proteomics. *Molecular systems biology*, 16(1):e9111, 2020.

Richard Novak, Miles Ingram, Susan Clauson, Debarun Das, Aaron Delahanty, Anna Herland, Ben M Maoz, Sauveur SF Jeanty, Mahadevabharath R Somayaji, Morgan Burt, et al. A robotic platform for fluidically-linked human body-on-chips experimentation. *Nature biomedical engineering*, 4(4):407, 2020.

Koji Ochiai, Naohiro Motozawa, Motoki Terada, Takaaki Horinouchi, Tomohiro Masuda, Taku Kudo, Motohisa Kamei, Akitaka Tsujikawa, Kenji Matsukuma, Tohru Natsume, et al. A variable scheduling maintenance culture platform for mammalian cells. *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, 26(2):209–217, 2021.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Essa Ghareeb, Justin Booth, and Samuel G Rodriques. Bioplanner: automatic evaluation of llms on protocol planning in biology. *arXiv preprint arXiv:2310.10632*, 2023.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2025.

Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B Tenenbaum, Leslie Kaelbling, and Michael Katz. Generalized planning in pddl domains with pretrained large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 20256–20264, 2024.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2998–3009, 2023.

Shodai Taguchi, Yasuyuki Suda, Kenji Irie, and Haruka Ozaki. Automation of yeast spot assays using an affordable liquid handling robot. *SLAS technology*, 28(2):55–62, 2023.

Carlos A Tristan, Pinar Ormanoglu, Jaroslav Slamecka, Claire Malley, Pei-Hsuan Chu, Vukasin M Jovanovic, Yeliz Gedik, Yogita Jethmalani, Charles Bonney, Elena Barnaeva, et al. Robotic high-throughput biomanufacturing and functional differentiation of human pluripotent stem cells. *Stem Cell Reports*, 16(12):3076–3092, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *IEEE Robotics and Automation Letters*, 2024.

Chengrui Wang, Qingqing Long, Meng Xiao, Xunxin Cai, Chengjun Wu, Zhen Meng, Xuezhi Wang, and Yuanchun Zhou. Biorag: A rag-llm framework for biological question reasoning. *arXiv preprint arXiv:2408.01107*, 2024.

Michael Wooldridge and Nicholas Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 6 1995. doi: 10.1017/s0269888900008122.

Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.

Nozomu Yachie and Tohru Natsume. Robotic crowd biology with maholo labdroids. *Nature biotechnology*, 35(4):310–312, 2017.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Junbo Zhang, Weiwei Wan, Nobuyuki Tanaka, Miki Fujita, Koichi Takahashi, and Kensuke Harada. Integrating a pipette into a robot manipulator with uncalibrated vision and tcp for liquid handling. *IEEE Transactions on Automation Science and Engineering*, 21(4):5503–5522, 2023.

Qiang Zhang, Keyan Ding, Tianwen Lv, Xinda Wang, Qingyu Yin, Yiwen Zhang, Jing Yu, Yuhao Wang, Xiaotong Li, Zhuoyi Xiang, et al. Scientific large language models: A survey on biological & chemical domains. *ACM Computing Surveys*, 57(6):1–38, 2025.

Zhiling Zheng, Oufan Zhang, Ha L Nguyen, Nakul Rampal, Ali H Alawadhi, Zichao Rong, Teresa Head-Gordon, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. Chatgpt research group for optimizing the crystallinity of mofs and cofs. *ACS Central Science*, 9(11):2161–2170, 2023.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.