**IFARLAB — Intelligent Factory and Robotics Laboratory**

Eskişehir Osmangazi University

# Language-Guided Pick-and-Place

A vision-language model for perception, measured geometry for the grasp

*Table 1: Scenario at a glance.*

| Item | Value |
|---|---|
| Task | Pick a part from the conveyor and place it in a bin, from a plain-language instruction |
| Robot | UR10e on a Festo linear axis (single-arm scenario) |
| End effector | OnRobot VGC10 vacuum gripper, mounted on the free slot of an OnRobot Dual Quick Changer |
| Sensor | Wrist-mounted SICK Visionary-T Mini — depth only, no colour channel |
| Reasoning model | Gemini Robotics-ER: returns a 2D point and text; it produces no robot actions |
| Measured cycle | 130 s mean over nine recorded runs on the physical cell |
| Maturity | End-to-end on the physical cell; suitable for look–plan–move, not for closed-loop servoing |

## 1. What this use case demonstrates

An operator types one sentence — for example “pick the part on the conveyor and put it in an empty bin on the top level of the rack”. No part model is taught, no grasp pose is programmed, and no fixture guarantees where the part lies. The cell finds the part, decides how to approach it, grasps it with vacuum and places it.

The architectural point of the use case is the division of labour. The vision-language model answers two questions: what the sentence means, and where in the image the object is. It returns text and a single two-dimensional point. Everything that determines whether the robot physically succeeds — the surface normal, the approach direction, reachability, collision avoidance, the motion itself, the grasp verification — is produced by measurement and by classical robotics code in the cell. This is a deliberate boundary: the model is used where it generalises, and metrology is used where it must be exact.

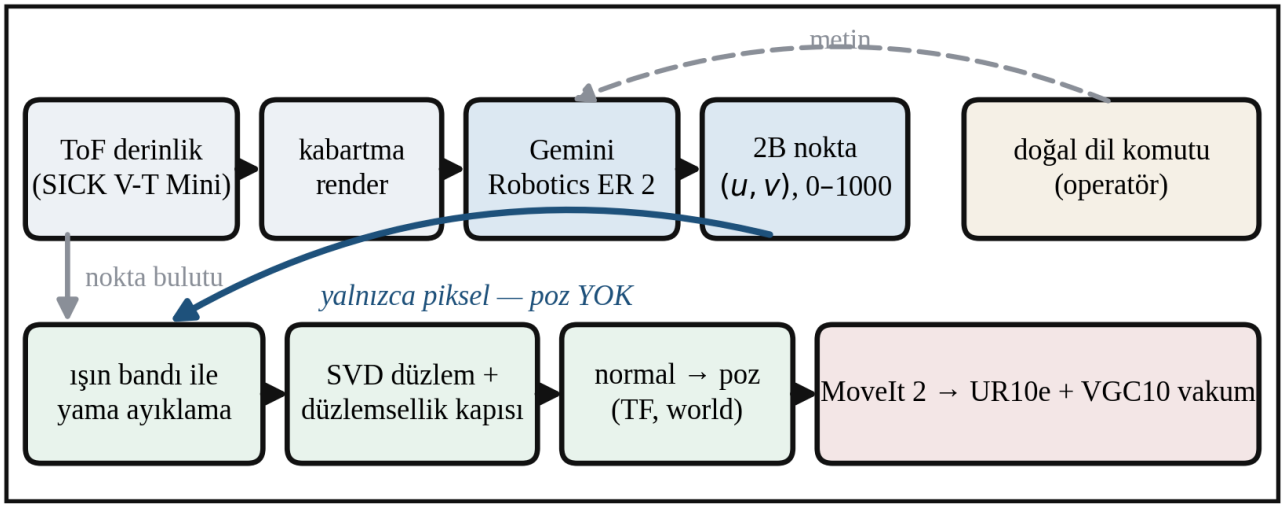



*Figure 1: Processing chain from the language command to the vacuum grasp pose. The upper row is what the model sees and produces; the lower row is the measurement chain that builds the pose — orientation never comes from the model.*

## 2. Cell configuration

*Table 2: Hardware used in this scenario.*

| Component | Specification |
|---|---|
| Robot | UR10e, six axes, with the linear axis available as a seventh joint |
| Tool changer | OnRobot Dual Quick Changer; the vacuum head occupies the free slot, so the scenario can be swapped with the parallel-gripper scenario without re-fixturing |
| Gripper | OnRobot VGC10 vacuum head, commanded over Modbus TCP; two independent channels, target vacuum 60 % (device limit 80 %) |
| Camera | SICK Visionary-T Mini time-of-flight, wrist mounted; publishes depth, infrared intensity and point cloud — there is no RGB stream |
| Conveyor | Belt stopped during scanning and grasping in the current scenario; measured belt surface height 0.845 m |
| Place target | Tool rack with four bin levels, measured at z = 0.938 / 0.727 / 0.517 / 0.259 m |

## 3. Why depth images rather than colour

The wrist camera has no colour mode. Rather than add a second sensor, the scene is rendered from depth into an image the model can reason about — and the same rendering is applied to simulated and real data, normalised over an identical metric range, so a behaviour validated in simulation transfers without changing modality.

The first rendering mode shaded surface normals. On a flat part lying on a flat belt this fails by construction: part and belt share the same normal, and the measured contrast between them was 1.7 grey levels out of 255. The mode now used renders height above the dominant fitted plane, which raises the same scene to 89.8 grey levels — a factor of 53, from the same depth data.



*Figure 2: Normal-shaded rendering: the 29 mm box is only a thin edge against the belt. Measured contrast 1.7 / 255.*

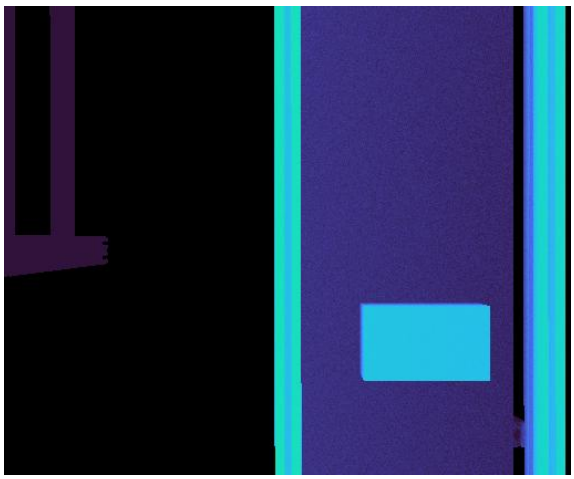


*Figure 3: Height-relief rendering of the identical frame: the box stands out as a clear rectangle. Measured contrast 89.8 / 255.*

The rendering mode is selected per scanning position, because the two views pose different problems: the top-down conveyor view is a height problem, while the side-on view of the rack saturates the height window and is better served by normals. Validation against known geometry: synthetic planes tilted from 0° to 55° with 5 mm of noise return the box at 29.0 mm and the floor at 0.0 mm, and a real frame returns 29.5 mm against a true 29 mm.

## 4. From a 2D point to a grasp

*Table 3: The pipeline and the gate at each stage.*

| Step | What the system does | Rejection condition |
|---|---|---|
| Scan | Move to a fixed scan pose, settle, render the depth frame | Pose unreachable or no depth stream |
| Interpret | Ask the model to split the sentence into a pick phrase and a place phrase | No pick phrase returned |
| Point | Ask the model to point at the target; deproject the pixel to 3D through the organised cloud | No depth or no transform at that pixel |
| Bound | Check the 3D point against the configured workspace volume | Outside the workspace box |
| Measure | Fit a plane to the depth patch around the point to obtain the surface normal | Fewer than 40 inliers, or plane residual above 4 mm — an edge or a corner |
| Verify | Solve inverse kinematics for a set of approach distances and for the contact pose | No feasible approach distance |
| Grasp | Cartesian descent, vacuum on, confirmation from the measured vacuum level | Vacuum below the confirmation threshold after two deeper attempts |
| Carry | Measure the part, attach it to the planning scene, re-validate the place target with the load | Measured size implausible, or place target invalid while loaded |
| Place | Descend, release, detach, retreat, return home | Short descent is tolerated, not fatal |

Each stage has an explicit rejection condition rather than a silent fallback. A grasp is never attempted on a surface the system could not measure, and the place target is re-checked once the part is in the gripper, because the attached load changes what is reachable.



*Figure 4: Contact: the suction cup seated on the top face of the part, at the orientation produced by the plane fit.*



*Figure 5: Lift-off 1.4 s after the vacuum is established; the part has been added to the planning scene as an attached body.*

## 5. Measured results

*Table 4: Nine recorded runs on the physical cell.*

| Metric | Measured value |
|---|---|
| Runs recorded on the physical cell | 9 |
| Runs that completed the full chain | 4 of 9 |
| Mean cycle time | 130 s (minimum 75 s, maximum 166 s) |
| Model waiting time per run | ≈ 38 s, about 29 % of the cycle |
| Mean latency per model call | 9.4 s (planning 7.4–7.6 s, pointing 4.6–16.4 s) |
| Vacuum establishment | 0.2 s; part clear of the belt 1.4 s after vacuum |
| Most frequent failure | Vacuum not established — the surface must be flat and non-porous |

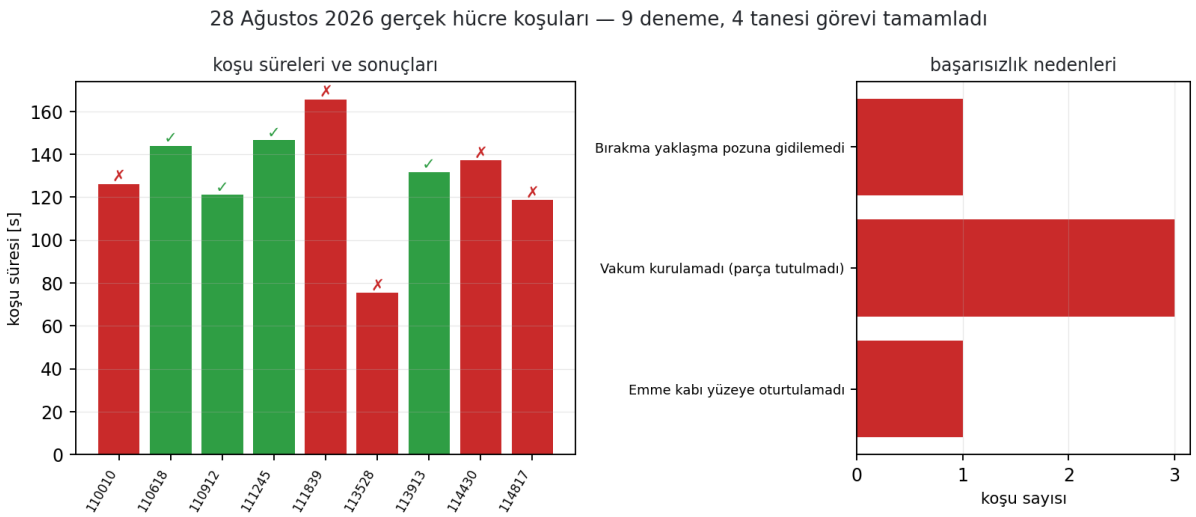



*Figure 6: Duration and outcome of the nine recorded runs, with the distribution of failure causes.*

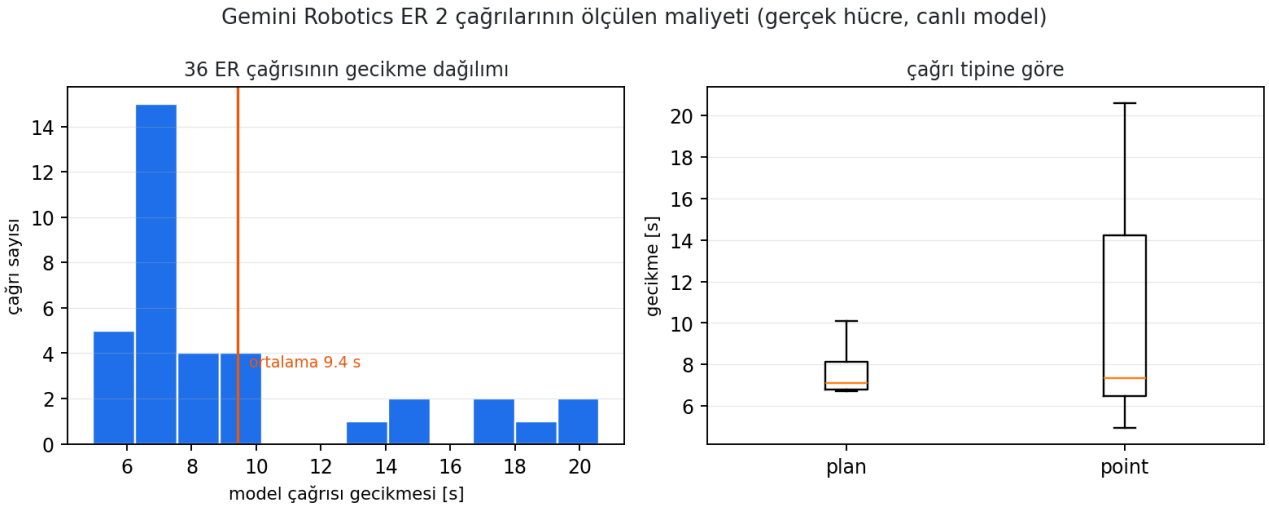



*Figure 7: Latency of the model calls, by call type. This is dead time in which the robot waits.*

## 6. Engineering findings worth reporting

- Measure the tool, do not trust the drawing. The suction cup geometry was fitted from the mesh: the suction axis is 30.00° off the frame X axis and the cup mouth sits 84.2 mm from the link origin. The previously entered value of 156 mm placed the commanded point 71.8 mm above every target. After the correction, the perpendicularity error went from 90° to 0.00° and the distance error from 71.8 mm to 0.04 mm.
- Depth conventions must be verified on the real sensor. The raw depth channel was found to be in 0.25 mm units and radial rather than planar; interpreting it correctly raised the share of valid pixels in the working window from 7.1 % to 99.1 % and removed an apparent 0.6 m of false doming at 1.8 m distance.
- A short inverse-kinematics budget costs nothing and saves a great deal. Reducing the solver timeout from 3.0 s to 0.5 s produced identical answers across 55 poses while cutting the total from 63.6 s to 1.7 s, because an infeasible pose consumes the full budget.
- A measured payload protects the planner. The part is measured before it is attached to the planning scene; one run measured the conveyor itself, which is why an upper size bound exists.
- Model latency, not motion, dominates the dead time. Roughly 29 % of a cycle is spent waiting for the reasoning model, which is acceptable for a look–plan–move cycle and rules out closed-loop visual servoing with this architecture.

## 7. Current limitations

- The grasp depends on vacuum, so the target surface must be flat and non-porous; this is the dominant failure mode in the recorded runs.
- There is no check that the release point lies inside the target rack. In two of the nine runs the model chose a placement back on the conveyor and the mission still reported success — a bounds check on the place pose is the first item on the roadmap.
- Each reasoning query is a cloud API call of several seconds, and the scenario does not start without a valid API key. The cell therefore depends on an external service for this scenario.
- The lateral calibration of the wrist camera mount has not been performed; only the optical offset along the view axis was measured.
- Simulation can validate perception, pose generation and motion, but not the grasp itself: the simulator does not model suction, so the part is attached by a joint instead.

## 8. Running the scenario

*The same scenario is reachable from the cell dashboard, which also provides the text box.*

```
# Simulation
ros2 launch gemini_robotics_ros gemini_pick_place.launch.py

# Physical cell, with run recording enabled
ros2 launch gemini_robotics_ros gemini_pick_place.launch.py mode:=real record:=true

# Issue the task
ros2 topic pub --once /gemini/command std_msgs/String \
  "{data: 'Pick the part on the conveyor and place it in an empty bin on the top level'}"
ros2 topic echo /gemini/status
```

**Every run is recorded.** A recording node writes the depth frames, the rendered images, the model queries and answers, joint states, vacuum signals and a full event timeline for each run. The numbers in Section 5 come from those recordings, and the same data is the basis for future learning-based work on the cell.