# Physical AI Leaderboard (PhAIL) White Paper

## Overview

**Physical AI Leaderboard (PhAIL)** is an ongoing, real-robot benchmark that aims to establish a **standard yardstick** for the new generation of physical-AI models. It measures model performance on commercially valuable tasks using production metrics such as throughput and reliability.

Current results are available at phail.ai.

## Introduction

Physical AI has moved fast over the last two years. Research results and marketing demos are everywhere, but **post-pilot, production deployments are still rare**.

Much of the current research and benchmarking effort focuses on home and personal-assistance tasks. At the same time, commercial settings – production and logistics – present a large and immediate opportunity where a single, reconfigurable system can handle many tasks and lower automation costs by reducing per-task setup time.

When we talk to operators in these industries, we hear the same questions:

**Is physical AI ready? Should we use it?**

**PhAIL** is designed to answer these questions with real-world evidence.

## Benchmark Design

For physical AI to see broad adoption, it will need to:

- **Make economic sense**: high throughput and high reliability (minimal human assists).
- **Be auditable**: runs, logs, and scoring that stakeholders can review.
- Be **fast to adapt** to a new task (low tuning effort and cost).
- **Generalize across tasks** (not a one-trick system).
- **Run on many embodiments** (no hardware lock-in; use what you already have).

These requirements guide PhAIL's design direction. The current version starts with a single embodiment and a single task; we intend to expand along all of these axes over time.

## Commercially valuable tasks

Many public tasks skew to home settings or one-off activities. In contrast, commercial operations favor repetitive work – exactly the kind of work people dislike, quality drifts over time, and attrition rises.

PhAIL validates models on **commercially valuable skills**. The first task is **bin-to-bin order picking** – where the robot moves items one at a time from an inbound tote to an outbound tote. Bin-to-bin picking is one of the most common operations in warehousing, fulfillment, and manufacturing, with millions of picks performed daily across these industries. It is physically straightforward, commercially important, and exposes exactly the throughput and reliability gaps that matter for deployment.

## Metrics

Academic work often reports **Success Rate**. It's useful, but businesses buy throughput and reliability.

PhAIL reports three metrics:

- **UPH** (Units Per Hour) – how fast the system works. Computed per object type as the total number of successfully moved items divided by the total evaluation time in hours. The headline UPH is the average across all object types, weighted equally – so each object contributes equally regardless of how many evaluation runs it has.
- **MTBF/A** (Mean Time Between Failures or Assists) – how long the robot runs before a human needs to step in. Computed as total evaluation time divided by the number of failed runs, expressed in minutes. A failed run is any run where the outcome is not "Success" (including safety stops, timeouts, and operator assists).
- **Completion** – the percentage of items successfully moved out of the total items placed in the inbound tote. Like UPH, computed per object type and then averaged equally across objects.

With current evaluation volumes (115-150 runs per model), pairwise differences in UPH are statistically significant when the gap exceeds approximately 15 UPH ($p<0.001$). Smaller differences may not be distinguishable at current sample sizes. The leaderboard is ongoing – statistical resolution increases as more runs are added.

All inference runs (videos and per-run logs) are publicly available for community review.

## Real hardware, not just simulation

Simulation makes resets and scoring easy, but buyers deploy in the real world. PhAIL runs on physical rigs with controlled procedures, so results reflect real perception, latency, and safety constraints.

## Hardware and sensors

PhAIL uses a **DROID-style setup**: a **Franka Research 3 robotic arm** with a **Robotiq 2F-85 gripper**. This configuration is widely available and reproducible (see DROID project for build instructions).

The station consists of an inbound tote (source) and an outbound tote (destination). The sensor setup includes two cameras:

- **Wrist camera** – mounted on the robot's end-effector, providing a close-up view of the grasp.
- **External camera** – mounted to the side of the workspace, providing a wider view of the station.

Between evaluation runs, the placement of the outbound tote (left or right side of the workspace) and the position of the external camera (left or right) are varied. This introduces spatial variability that tests whether models are robust to changes in workspace layout.

Camera position, tote placement, and object set are recorded in the run log.

## Fine-tuning dataset

Today's models do not reliably complete these tasks from text prompts alone – none of the models we have evaluated demonstrate zero-shot capability on this task. For the initial evaluation, fine-tuning was a prerequisite – no model we tested produced meaningful results without it. The goal of the fine-tuning dataset is not to maximize model performance, but to provide a standardized baseline so that all models are compared on the same data.

We collect teleoperated demonstrations using a VR interface and publish the dataset under a non-commercial license. The current dataset contains **352 episodes** (~12 GB) across four object types:

| Object | Episodes | Total duration |
|---|---|---|
| Wooden spoons | 167 | ~340 min |
| Towels | 112 | ~178 min |
| Scissors | 83 | ~160 min |
| Batteries | 88 | ~134 min |

We also publish open-source fine-tuning scripts. Participating teams can use these scripts to fine-tune their own models and submit the resulting checkpoints to the leaderboard.

## Evaluation variability

Real-world environments never stay the same. Lighting changes from day to day and season to season, cameras drift slightly, hardware wears, and product packaging evolves. A capable model must remain reliable when the conditions change.

To test robustness, PhAIL introduces controlled variation between evaluation runs:

- **Tote placement** – the outbound tote is placed on either the left or right side of the workspace. This changes the required reach and approach angle for the robot arm.
- **Camera position** – the external camera is mounted on either the left or right side. This changes which parts of the workspace are visible or occluded from the camera's perspective.
- **Blind model rotation** – the scheduler randomly selects which model checkpoint runs next. The operator does not know which model is being run, preventing unconscious bias.

These variations are balanced across runs, with roughly equal numbers of episodes for each configuration.

## Observations and actions

During inference, models receive:

- **Camera images** – full-resolution frames from both the wrist and external cameras. Models may resize these internally as needed.
- **Proprioception** – end-effector pose (position + orientation) and gripper state.
- **Language prompt** – "Pick all the items one by one from transparent tote and place them into the large grey tote."

Models output actions at **15 Hz** in one of several supported formats: absolute end-effector pose, absolute joint positions, or joint deltas. Gripper control is binary (open/close). The positronic library provides a codec abstraction layer that translates between model-specific action representations and the robot's control interface, so new entrants can use their preferred action space without modifying the evaluation infrastructure.

### Independence and governance

PhAIL is operated by Positronic Robotics. Positronic does not submit its own models to the leaderboard. All evaluation episodes – including failures and safety stops – are published in full, with synchronized video and telemetry, so that any result can be independently verified.

The benchmark is governed by an open consortium. Founding partners are Nebius (compute) and Toloka (data). The consortium is designed to ensure neutrality and broaden the benchmark's scope as it grows. Organizations interested in shaping the benchmark – whether as model submitters, hardware vendors, or evaluation partners – can reach out at hi@phail.ai.

## Evaluation protocol

This section defines how a submitted model checkpoint is validated on a single run in the bin-to-bin setup. It is written to be clear, unambiguous, and auditable.

### Scope and terms

- **Checkpoint** – a model artifact produced by the open-source fine-tuning scripts, or a complete model package submitted by a participant.
- **Inference host** – a local machine running the positronic library for action streaming.
- **Operation** – bin-to-bin order picking of a single SKU.
- **Unit** – one picked item moved from inbound to outbound.
- **Assist** – any human action that changes robot state, scene state, or task progress.
- **Time cap T** – maximum wall-clock time for the run, set to **30 seconds per item** (e.g., 4 minutes for 8 items). For reference, a human operator completes the same task at approximately 1,300 UPH (~2.7 seconds per item), so the 30-second cap provides roughly 10x the human pace.
- **Run** – one attempt to move N units under the time cap with a specific checkpoint.

## Software and hardware

- Inference runs locally via the positronic library.
- The station consists of an inbound tote and an outbound tote.
- Two cameras capture each run: a wrist-mounted camera and an external camera.
- Camera position, tote placement, and object set are recorded in the run log.

## Blinding and randomization

- The scheduler randomly selects which checkpoint runs next; the operator does not see model identity during the run.
- Object SKU and batch size for the run are drawn randomly and logged into run meta information.

## Pre-run setup

1. Home the robot and pass end-effector health checks.
2. Place the outbound tote on either the left or right side of the workspace (varied between runs).
3. Place the inbound container as per the station layout. Load items of one SKU.
4. Set the item count and tote/camera configuration in the evaluation interface.
5. Arm the run in the UI (this also starts synchronized video and telemetry logging).

## Run start

- The operator presses Start. The UI sends the task spec to the inference host, starts the timer (30 seconds × number of items), and enables model control.

## During the run

- The model controls the robot autonomously. The intended behaviour is to pick one item at a time from the inbound tote and release it inside the outbound tote. Multi-grasps (picking more than one item) are not penalized as long as all items end up inside the outbound tote.
- The operator does not touch the robot or the environment while the episode is running. The only intervention available is triggering a safety stop, which ends the run.

## Success criteria

**Per-unit success** – a unit is released fully inside the outbound container. This is verified by the operator at the end of the run.

**Per-run success** – all N units are completed within the time cap with no safety stop. When the operator observes that all items have been moved, they press Stop to end the run. The elapsed time therefore includes the operator's reaction time. This is consistent across all models because the operator is blinded to which model is running, and will be replaced by automated detection (scale-based) in a future version.

## Failure and end conditions

A run ends when any of the following occurs:

- **Timeout** – the time cap expires before all items are moved. The run is scored based on the number of items successfully moved before time expired.
- **Safety stop** – the operator triggers a safety stop when the robot behaves dangerously (e.g., attempting to place one tote inside another, pushing totes off the table, or making dangerous contact with the workspace). The run is marked as "Safety."
- **Irrecoverable controller error** – the run is marked as failed.

Note: when the robot's built-in safety reflexes trigger (e.g., force/torque limits), the system automatically resets the arm, clears queued actions, and resumes inference. This does not end the run – it is part of normal recovery behaviour.

## Timing

- Wall-clock timing only. The time cap is **30 seconds per item**. Pauses are not allowed; a safety stop ends the run.

## Scoring and artifacts

For each run we compute:

- Units successfully moved, total items in the inbound tote, elapsed time, UPH, completion percentage, and outcome (Success / Fail / Safety).

We store and publish:

- Synchronized multi-camera video, robot state and action logs, tote/camera configuration, and all scoring metadata.

All evaluation data is publicly downloadable as a Positronic dataset.

### Reset between runs

1. Return to home using the reset macro.
2. Empty the outbound container and reload inbound for the next run.
3. Verify logging is active, then arm the next run.

## Future work

PhAIL is designed to grow along several axes:

- **Unseen objects** – evaluating models on object types not present in the training dataset, to measure true generalization rather than task-specific memorization.
- **Additional tasks** – insertion (e.g., USB-A, Ethernet connectors) and additional pick-and-place variants beyond bin-to-bin.
- **Additional embodiments** – expanding beyond the current Franka FR3 setup to other robot arms and mobile manipulators, testing whether models generalize across hardware.
- **Environment generalization** – evaluating in workspace setups not seen during training (different rooms, lighting, backgrounds).
- **Automated success detection** – adding scales under the outbound tote to detect per-item placement and automatic run completion, removing the need for operator-triggered episode termination.