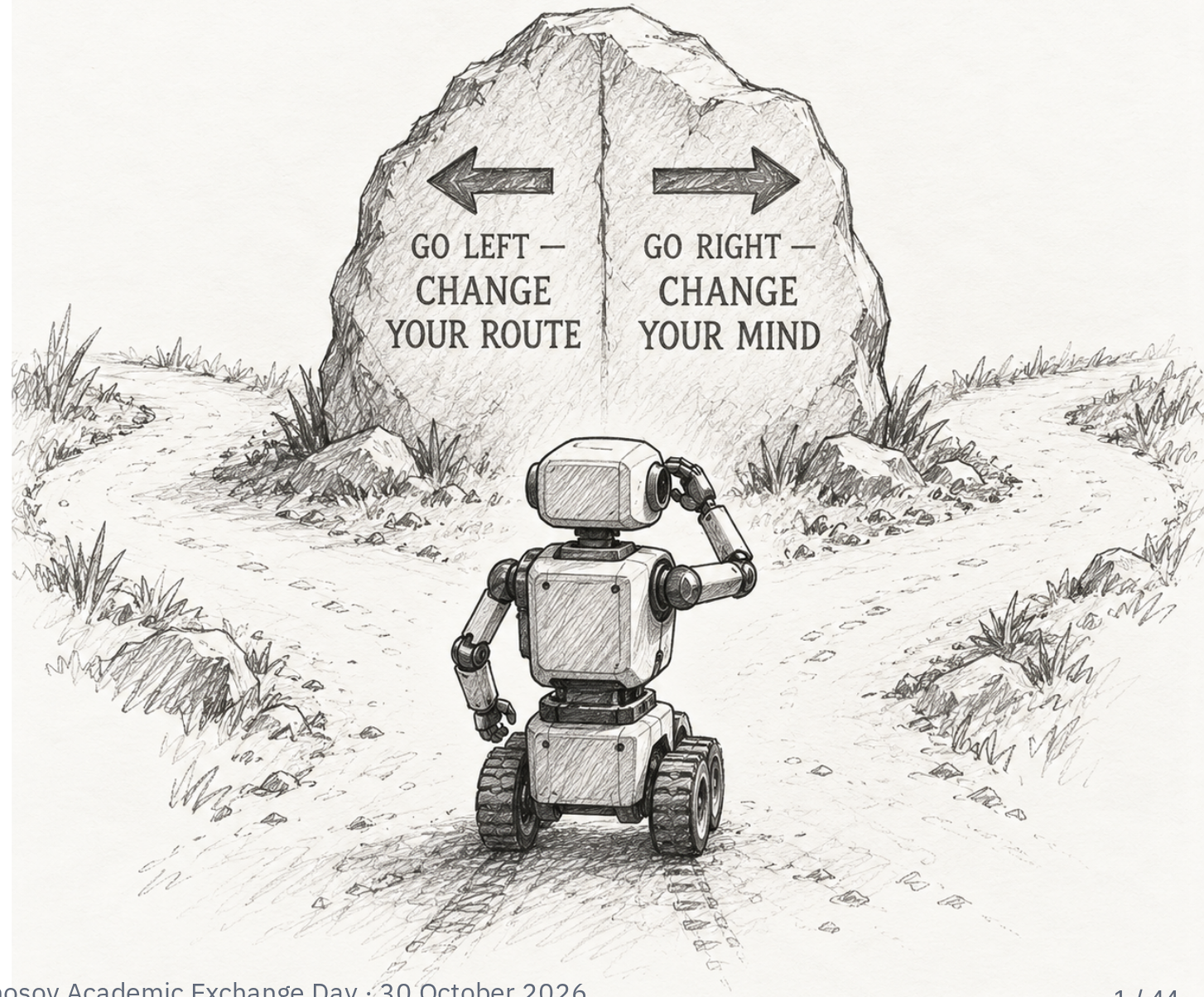
Long Paths and Shortcuts to Understanding VLA

Experiments in Planning, Reward Design, and Adaptation to a New Robot

Alexei Kornaev, Dr.Sc., Prof.

Innopolis University

Physical AI & VLM
Lomonosov Academic Exchange Day
30 October 2026 · Moscow



Our path to embodied robotics

Reproduce existing VLA systems

Within available compute

→

Test on our tasks

Limited perception
Complex goals
Changed embodiment

→

Physical understanding + uncertainty

Future research:
physics-informed methods
PINN/iPINN · JEPA
knowing when to ask

→

Embodied robotics

Long-term ambition

↓

Set the robots free

↓

Let them work hard

↓

Let them hand over their salaries to us

↓

Let them take care of us and the environment

Our current focus

High-level planning

Guide a fixed VLA executor without retraining it

Reinforcement learning

Adapt the policy in simulation to a new robot: CDPR

The people behind the experiments



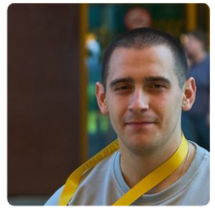
Alexei Kornaev
Scientific supervisor



Alexander Maloletov
Head, NTI Center*



Damir Nurtdinov
VLA / RL, CDPR



Denis Degtyarev
Planner evaluation



Zlata Mishustova
Wheeled robot



Kamil Nuriev
Rewards, physical criteria



Egor Novikov
VLA adaptation, rewards



Artem Tuzov
Data and tooling



Yuri Kazakov
Cartography, control

*NTI Center for Robotics and Mechatronics Component Technologies

Part 1 · Planning above a fixed controller

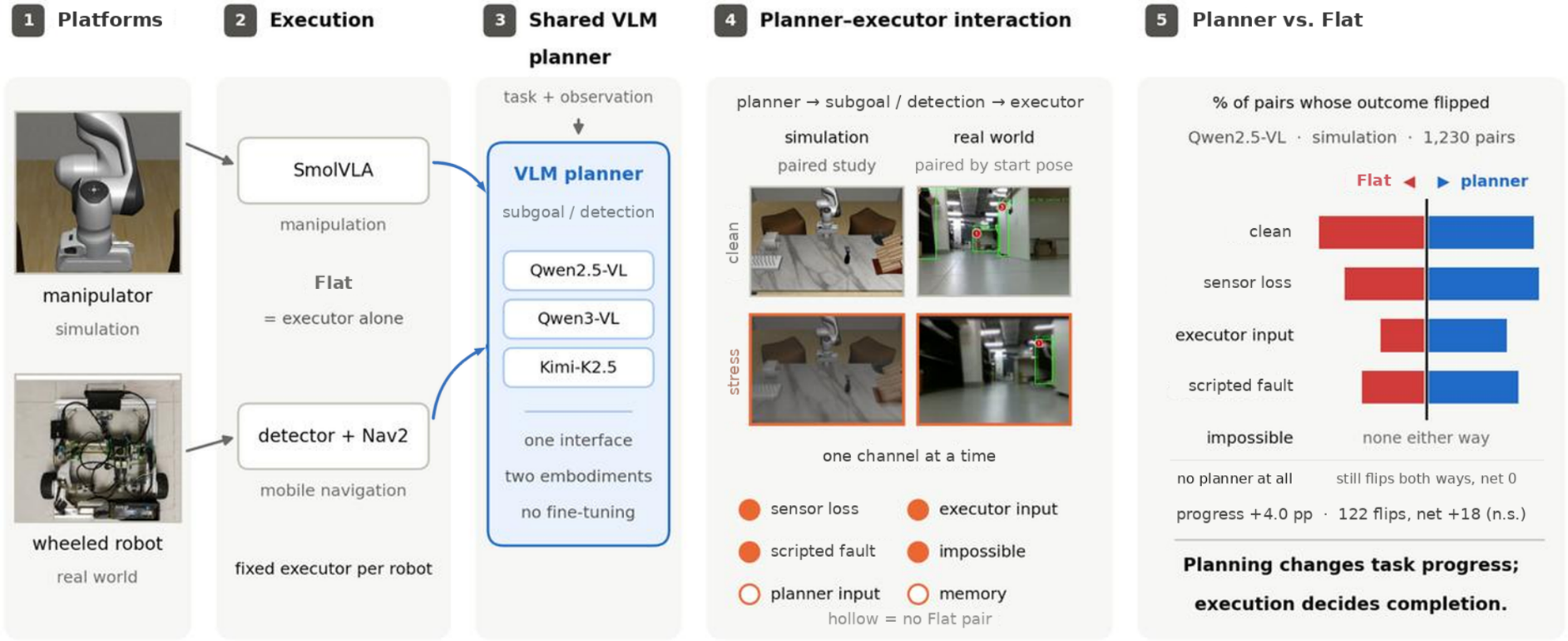
01
Planning above a fixed controller

02
Learning for a new embodiment

03
What comes next

Can better guidance help a controller that we do not retrain?

Planner pipeline: manipulation



Manipulation study: planner-selected subgoals for a fixed SmolVLA executor.

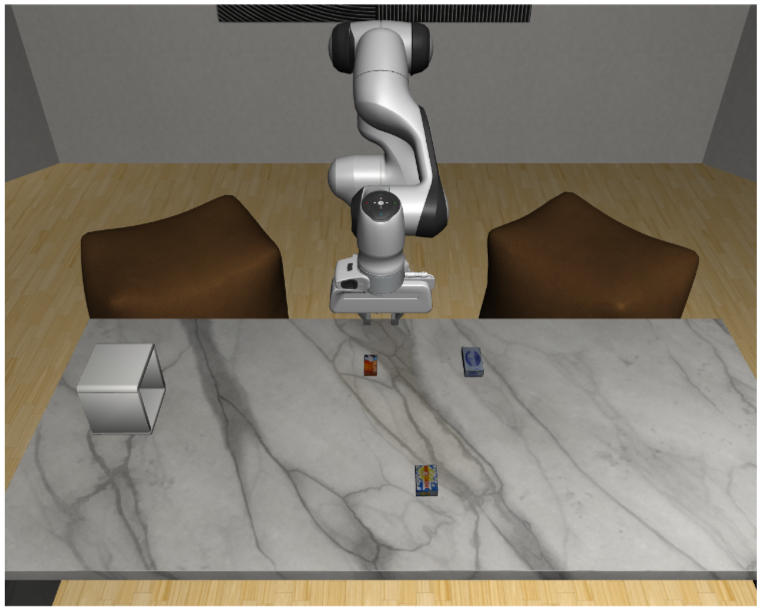
Our contribution: measuring help, harm, and chance

Count what a net success rate hides.
Replay each task from the same start, with and without guidance; count rescued failures and disrupted successes separately.

Measure how much change happens without a planner.
Replay the fixed policy against itself to estimate its own outcome variability.

Test the cost and limits of intervention.
Compare planner models, reference goals and larger execution budgets to examine why progress may improve while completion does not.

What can the planner change?



“Put the three food boxes into the organizer.”

Planner

Instruction + observation

↓ current goal

“Pick up the butter.”

↓

Fixed SmolVLA

Images + state + goal → actions

Pick / place cream cheese → Pick / place popcorn → Pick / place butter

The planner selects the request; the executor performs the action.

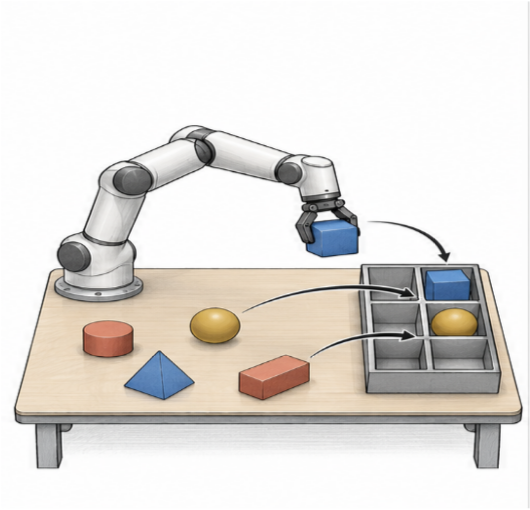
SayCan · GRAPPA · RoboCerebra · SmolVLA

What did we test?

Fixed SmolVLA executor · three planner models · 1,230 matched episodes per planner

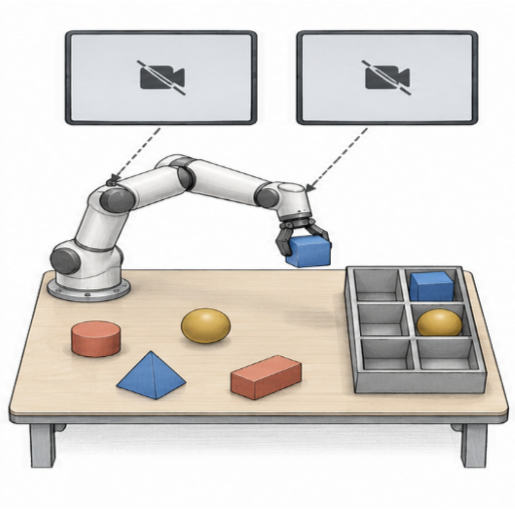
Normal operation

Long, multi-step tasks



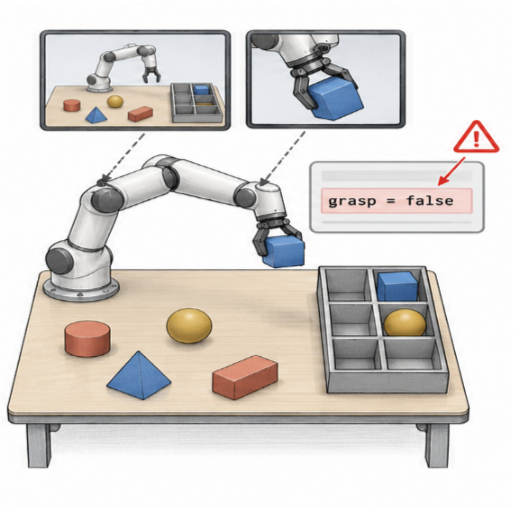
Less visual input

Camera information removed



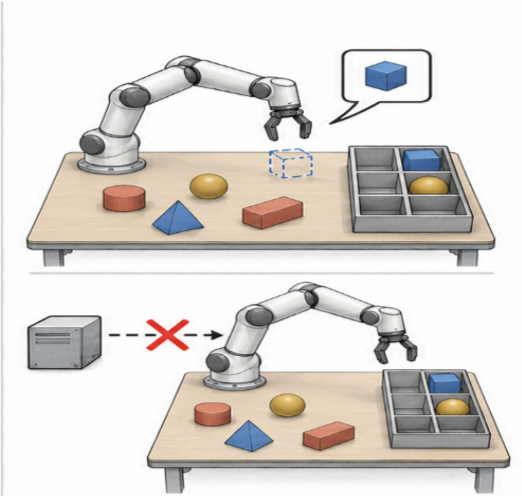
Reported faults

Predicate signal changed



Impossible tasks

Missing targets or blocked actions



Segment-based evaluation: state restored at subgoal boundaries. Reported faults alter the monitor signal.

Compare the same starting state

The same success rate can hide different successful episodes.

Restore the same initial state

Flat

Complete instruction

Fixed executor

Planner

Intermediate goals

Same executor

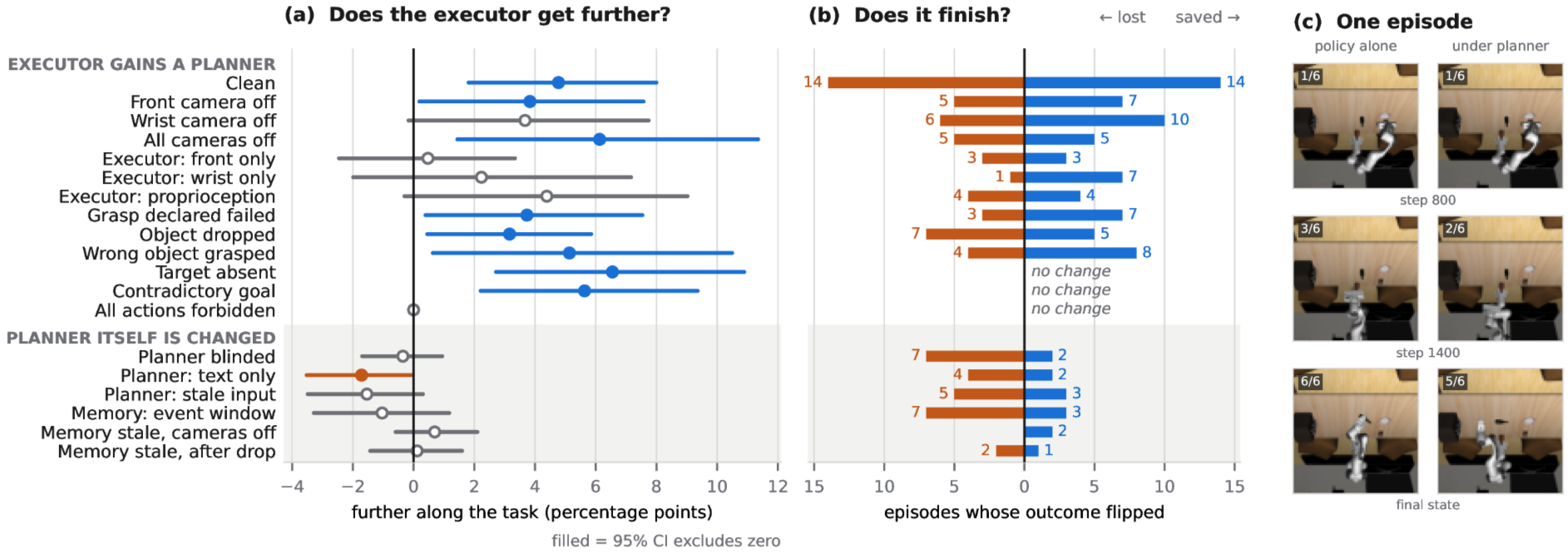
Same task, start, stress condition and control budget

Flat	Planner	Outcome
Success	Success	Both succeed
Failure	Success	**Rescued**
Success	Failure	**Newly failed**
Failure	Failure	Both fail

Initial-state pairing and later subgoal-boundary restoration are separate operations.

Manipulation: progress and changed outcomes

Qwen2.5-VL: +4.0 progress points · 70 rescued / 52 newly failed · net +18 completions (p = .12).



Upper block: 1,230 matched pairs, 13 conditions. Shaded rows use their own baselines.

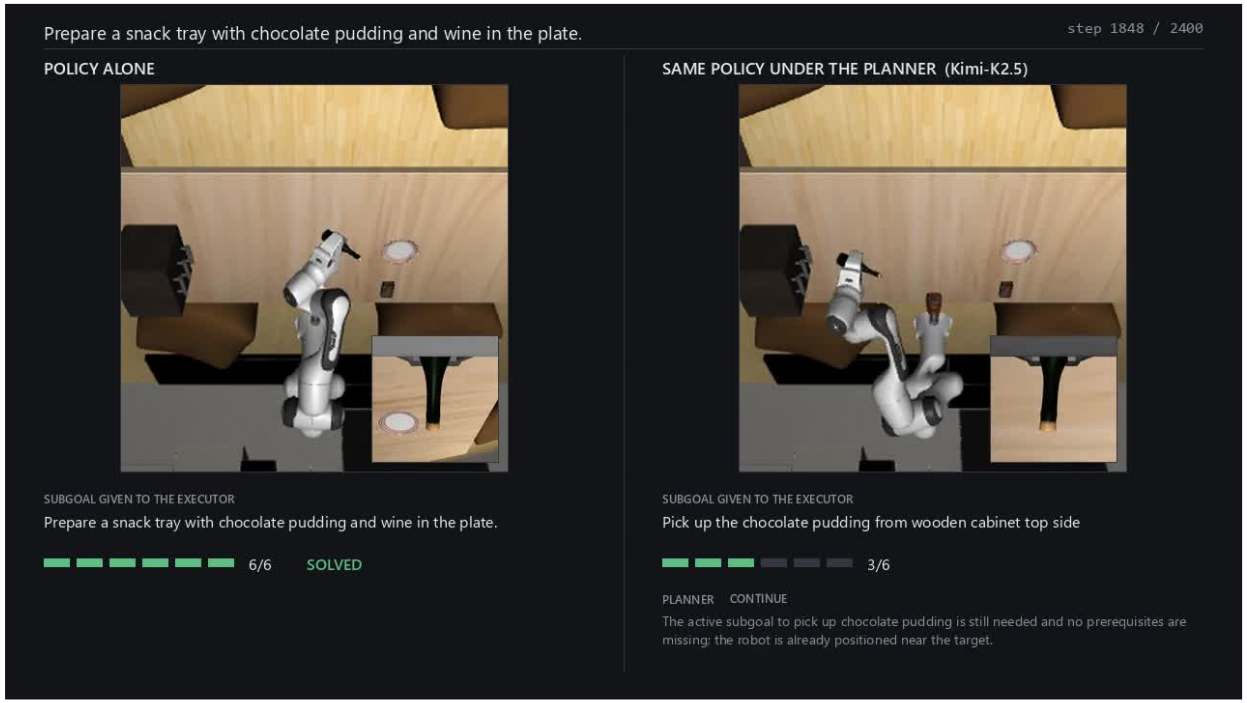
Previously successful episodes can fail under guidance

Among the 176 episodes solved by the fixed policy:

Goal source	Successes lost	Conditional loss rate
Qwen2.5-VL	52 / 176	29.5%
Qwen3	56 / 176	31.8%
Reference goals	76 / 176	43.2%
Kimi-K2.5	77 / 176	43.8%

Manipulation: a paired execution demonstration

Left: Flat model (policy alone) **Right: Planner-guided policy**



Flat on the left · Planner on the right · original playback speed

Kimi-K2.5 planner · shared execution clock · fixed policy in both branches.

Almost finished is still unfinished

Policy alone



With planner



Final state under the same control budget

One matched episode

6 / 6
Policy alone

5 / 6
With planner

The guided policy gets close,
but the task is incomplete.

Kimi-K2.5 · Ideal/case5/seed0
Different from the Qwen headline experiment.

Count completed tasks as well as partial progress.

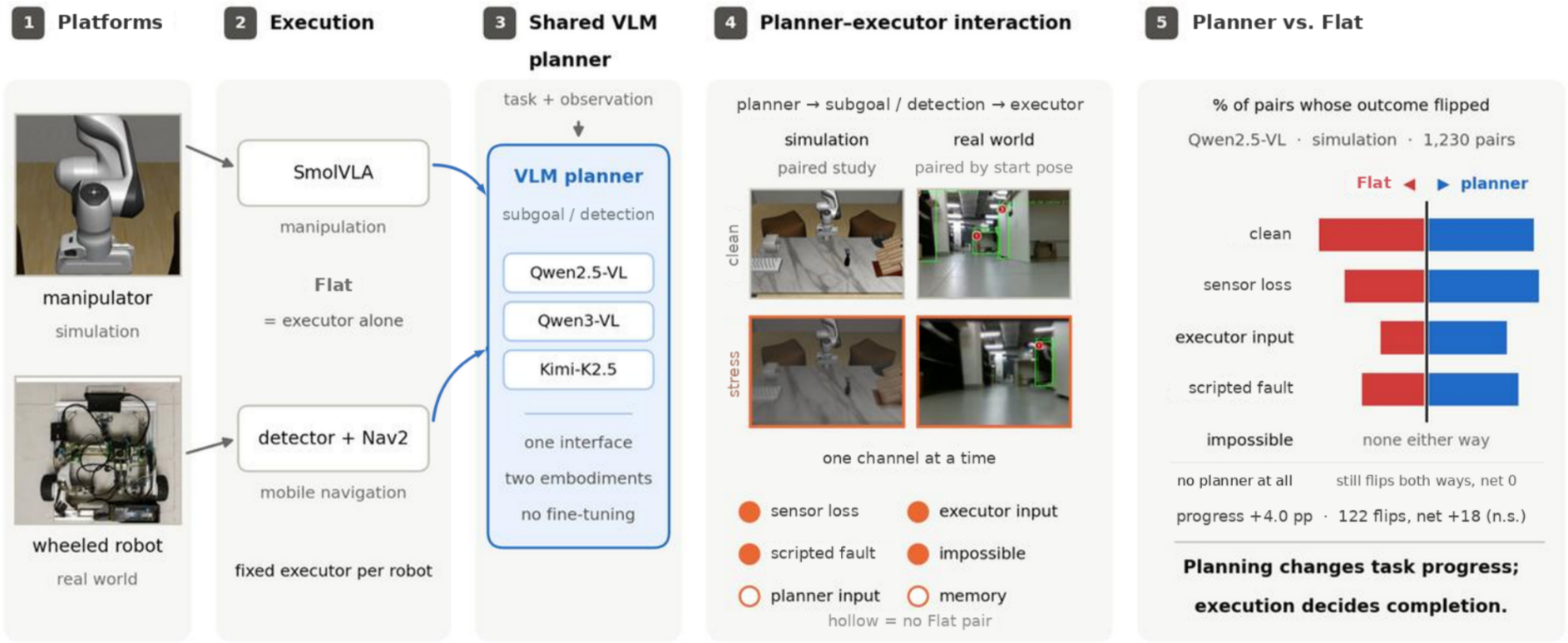
Manipulation study: findings

More progress did not clearly translate into more finished tasks.
Guidance helped intermediate steps, but the overall completion benefit remained uncertain.

Guidance could help one attempt and disrupt another.
Some tasks that succeeded with the policy alone failed after planner intervention.

Useful guidance still depended on execution.
Reference subgoals did not ensure completion; extra time did not improve Qwen3 completion in the tested comparison.

Planner pipeline: wheeled-robot navigation



Second study: object search with a fixed detector and Nav2 navigation stack.

Wheeled robot: task and comparison

Can planner guidance help the robot reach the requested object when the baseline cannot construct or retain a usable target?

Flat system

Fixed search, detection and approach sequence.

The robot must reach the designated object, not just another object of the same class.

Planner-enabled system

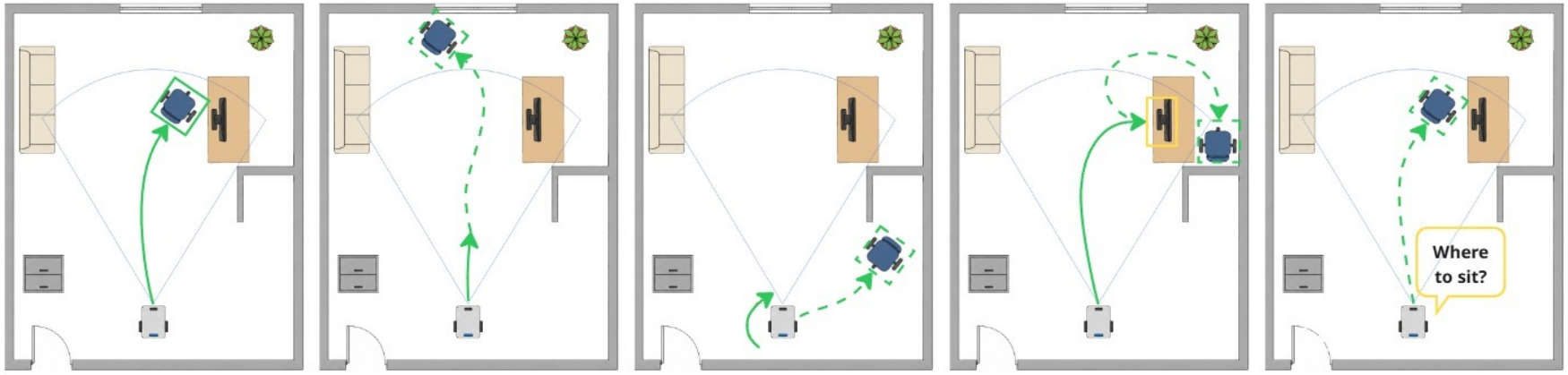
Qwen3-VL selects navigation skills and uses target context and memory.

Same starting pose and heading; five scenes, five trials per scene and system.

Fixed detector and Nav2 executor. This evaluates the broader planner-enabled stack, including memory and interpretation.

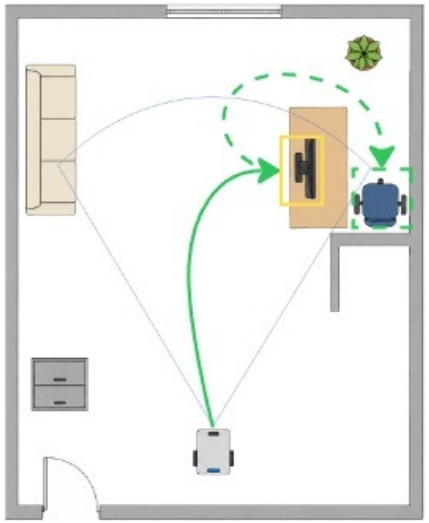
Mobile navigation: scene-level completion

System	1 Visible and reachable	2 Fixed scan	3 Outside the map	4 Around the corner	5 Implicit description	Total
Flat	5/5	5/5	1/5	2/5	0/5	13/25
Planner	5/5	5/5	5/5	4/5	5/5	24/25



Five reused scenes × five trials; manual, unblinded scoring. Comparison includes memory and semantic interpretation.

Guidance helped recover a usable target



Target localized → approach blocked
Reposition → retry the remembered target

13/25
Flat

24/25
Planner

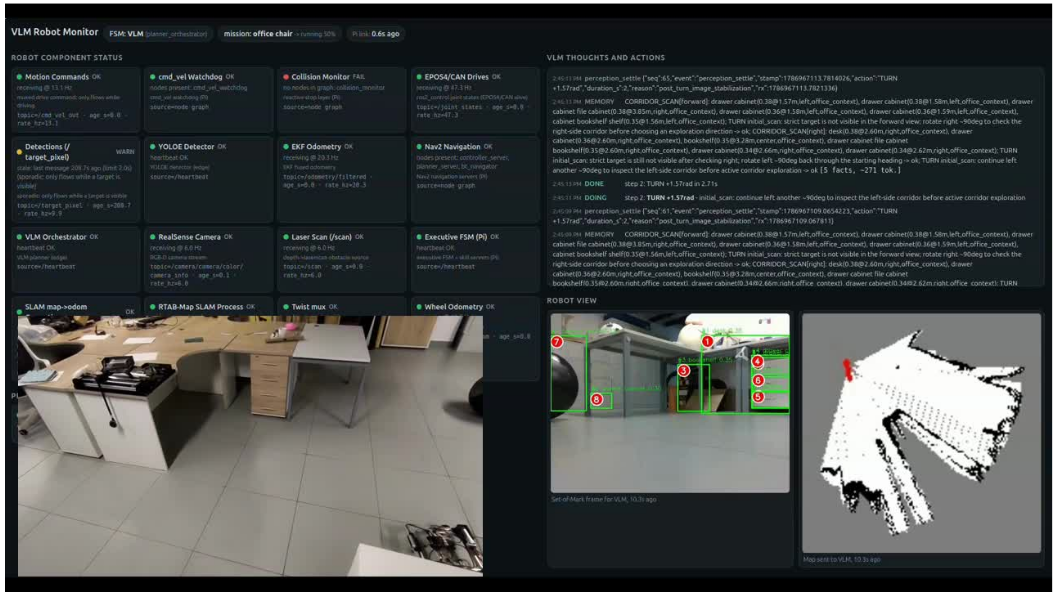
Target	Flat	Planner
Already usable	10/10	10/10
Needs memory or interpretation	3/15	**14/15**

Success: reached the designated object.

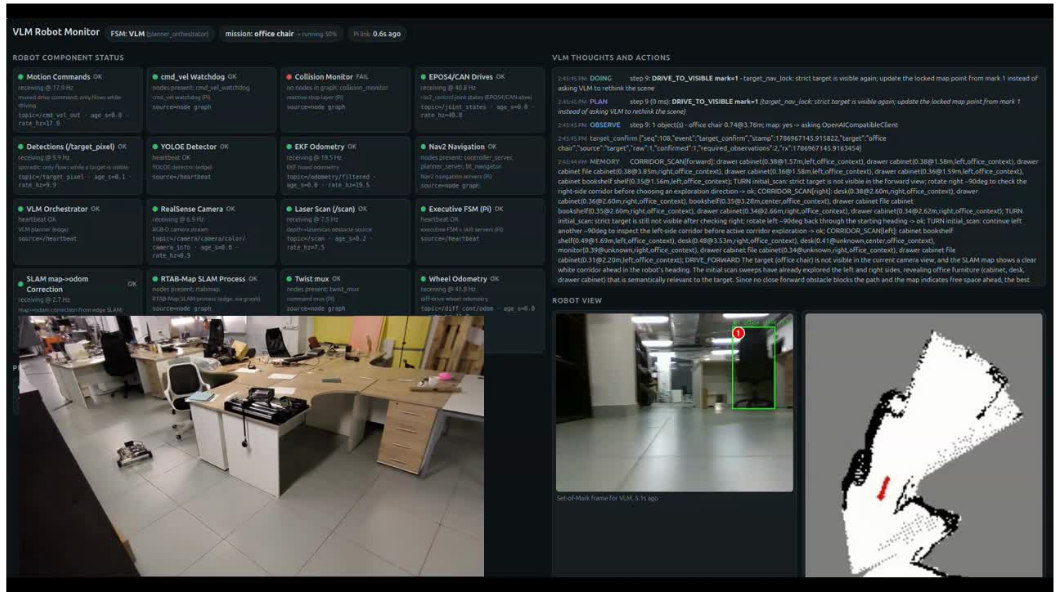
Benefit concentrated where the baseline lacked a usable target.

Physical mobile robot · detector + Nav2 · five scenes × five trials · unblinded scoring

Mobile navigation: observation and execution



First half: supplied mobile-robot demonstration · 2× speed



Second half: continued demonstration · 2× speed

Interpret the requested target, retain useful observations, and navigate using the detector and Nav2 stack.

Wheeled-robot study: findings

The planner helped when the target was hard to identify or remember.
Context and memory helped continue the search and approach.

Simple cases already worked with the baseline.
When the target was visible and usable, extra guidance added little.

The benefit came from the whole system.
Planning, memory and interpretation worked with the fixed detector and navigator.

The evidence is limited to the tested scenes.
Broader reliability still needs evaluation.

Part 2 · Learning for a new embodiment

01
Planning above a fixed controller

02
Learning for a new embodiment

03
What comes next

How can a robot learn new actions from its attempts—and keep its earlier skills?

From the base policy to our reward experiment

OpenVLA-OFT

Images + state + instruction → robot actions
Our starting policy in LIBERO

Starting action policy

SimpleVLA-RL

Collect simulated attempts;
update the policy using rewards

Rollout-and-update framework

Policy
Images + state + instruction

→

Rollouts
Attempts at the task

→

Reward
Score progress or success

→

RL update
Improve the policy

OpenVLA-OFT gives us the policy; we test how reward design changes its learning.

Training feedback = numerical reward · CDPR uses a different policy: SmolVLA

Kim et al., OpenVLA-OFT (2025) · Li et al., SimpleVLA-RL (2025)

OpenVLA-OFT: the policy we build on

Inputs

Camera observations, language instruction and robot state.

Action prediction

Continuous actions predicted in chunks with parallel decoding.

Optimized fine-tuning

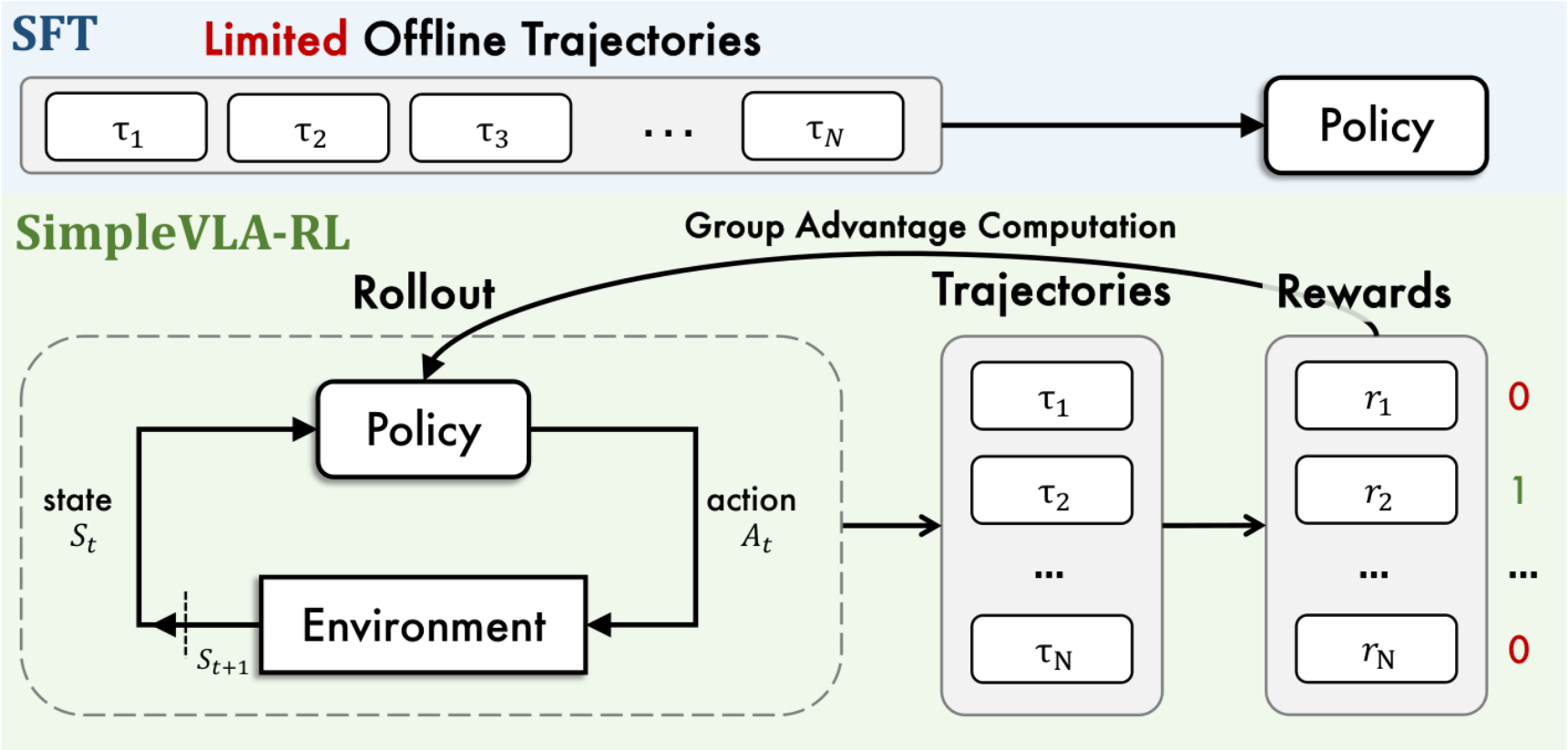
The OFT recipe studies action representation, decoding and learning objectives.

Our use

An OpenVLA-OFT-based policy, adapted with LoRA in the LIBERO reward study.

Kim, Finn & Liang (2025), Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. https://arxiv.org/abs/2502.19645

SimpleVLA-RL: learning from simulated rollouts



Sample trajectories, score outcomes, and update the policy from relative performance.

Li et al. (2025), SimpleVLA-RL, Figure 2. https://arxiv.org/abs/2509.09674

LIBERO reward comparison

Published starting point

OpenVLA-OFT policy
SimpleVLA-RL rollout-and-update recipe
LoRA fine-tuning in simulation

Ten LIBERO-Spatial manipulation tasks.

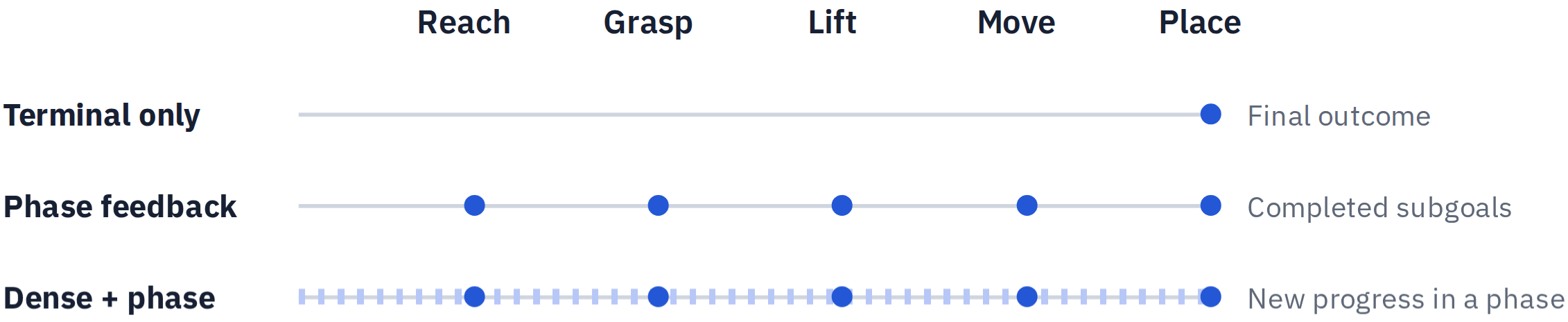
Our reward experiment

Same 61.0% initialization
160 updates: terminal-only and dense + phase
Terminal verifier coefficient: 5

300 validation rollouts per endpoint.

Train the manipulator policy; vary reward feedback; keep the native success test.

Reward new progress, not repeated motion



Within one phase

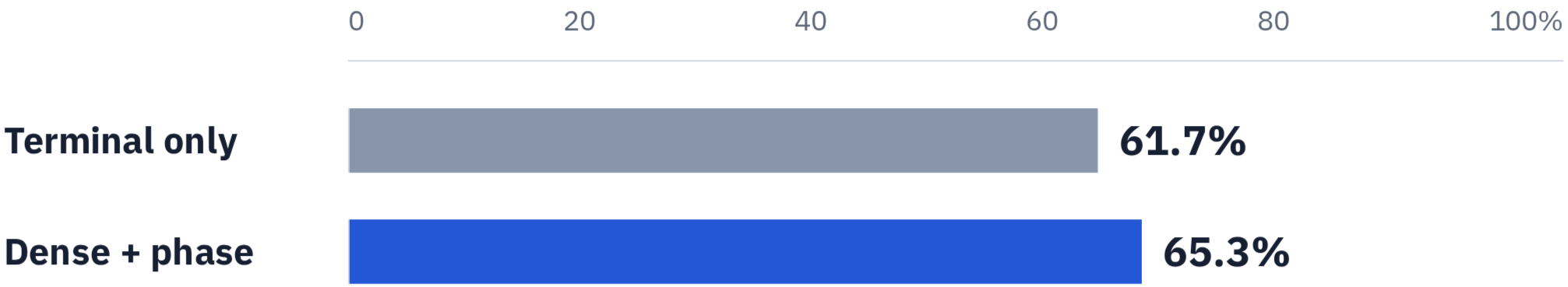
Progress	**0.2**	**0.6**	**0.3**	**0.6**	**0.8**
New credit	+0.2	+0.4	0	0	+0.2

No extra progress credit for returning to an old maximum.

Our LIBERO reward method. Unweighted illustrative increments; progress baseline resets at each new phase. Success verifier unchanged.

Dense + phase feedback improved the endpoint

LIBERO-Spatial · OpenVLA-OFT-based policy · 160 updates each



Common initialization: **61.0%** · terminal reward weight: 5
300 validation rollouts per endpoint

+3.6 percentage points between final endpoints

Descriptive endpoint gain; independent training-seed uncertainty is not established.

Intermediate feedback: benefit and design cost

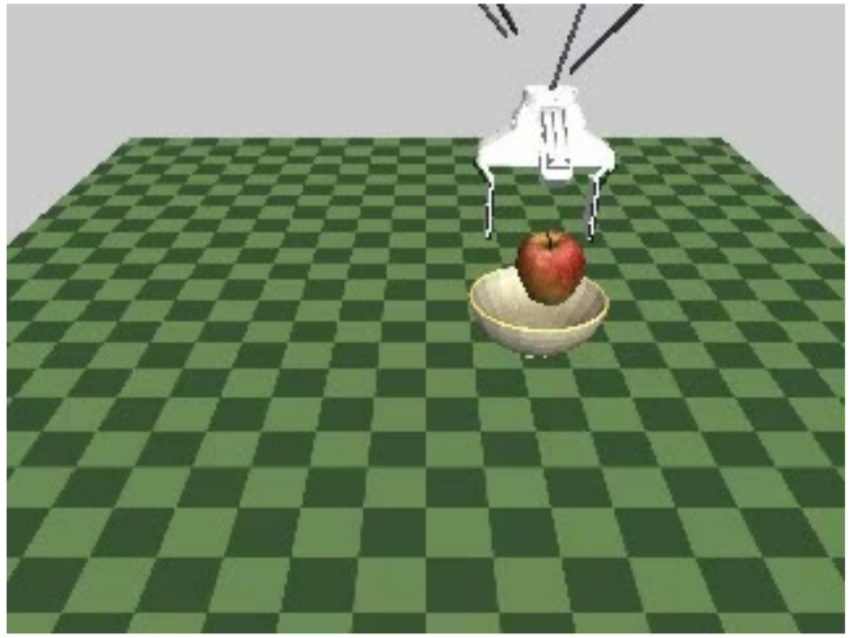
Feedback	Updates	Success
Terminal only	160	61.7%
Dense + phase	160	65.3%
Phase only	110	62.3%

Dense shaping rewards new progress: $\Delta p_t = \max(p_t - p_{best}, 0)$.

Dense + phase reached a higher endpoint here. Its progress functions require task-specific design; binary milestones need completion predicates.

Terminal coefficient: 5 for terminal-only and dense + phase; 1 for phase-only. Phase-only also uses fewer updates.

A cable robot changes the control problem



Cable-driven parallel robot (CDPR) · simulation

“Put the apple into the bowl.”

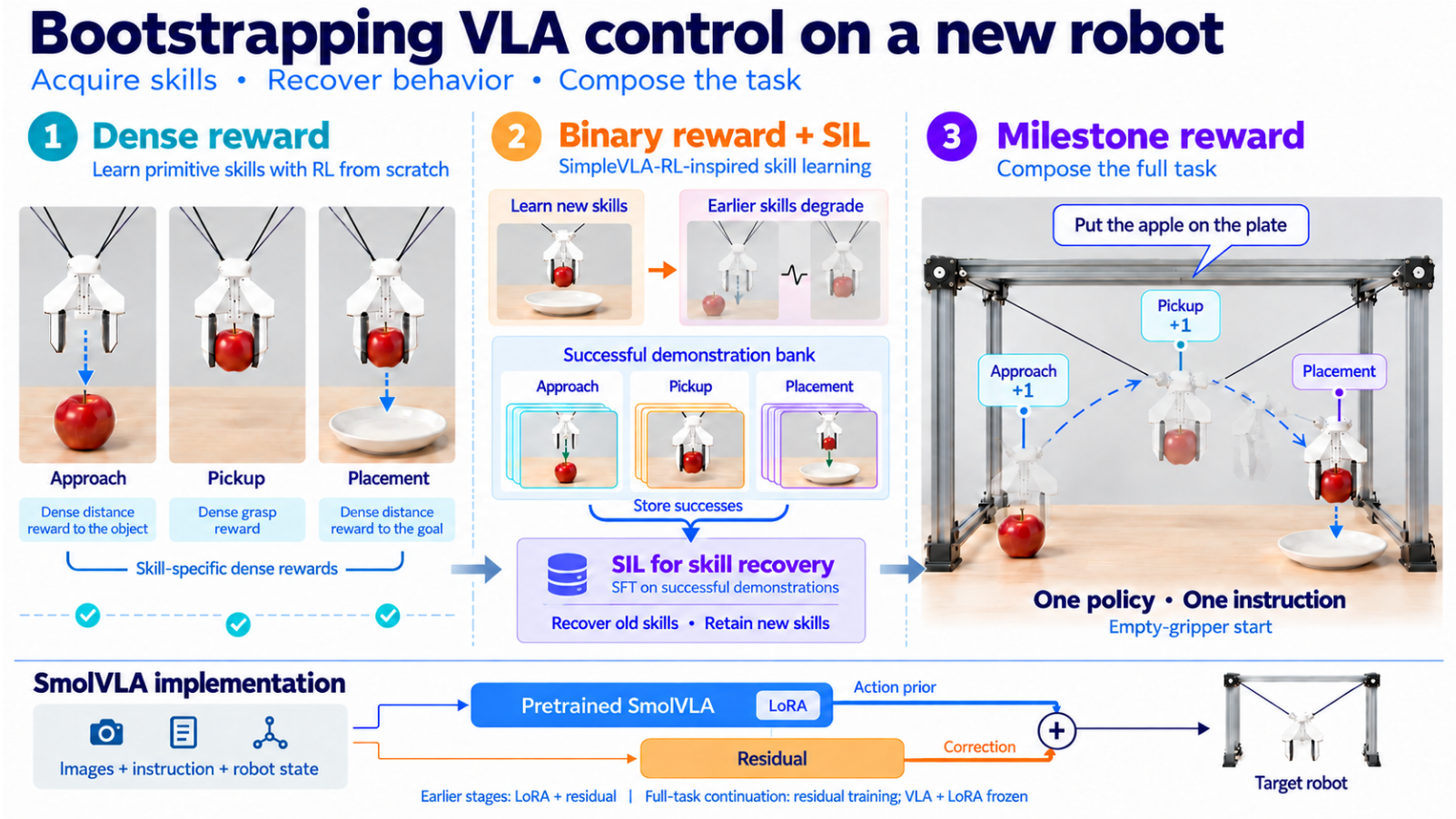
Five policy channels

Translation: x, y, z
Yaw
Gripper

Fixed inverse kinematics maps
Cartesian targets to cable lengths.

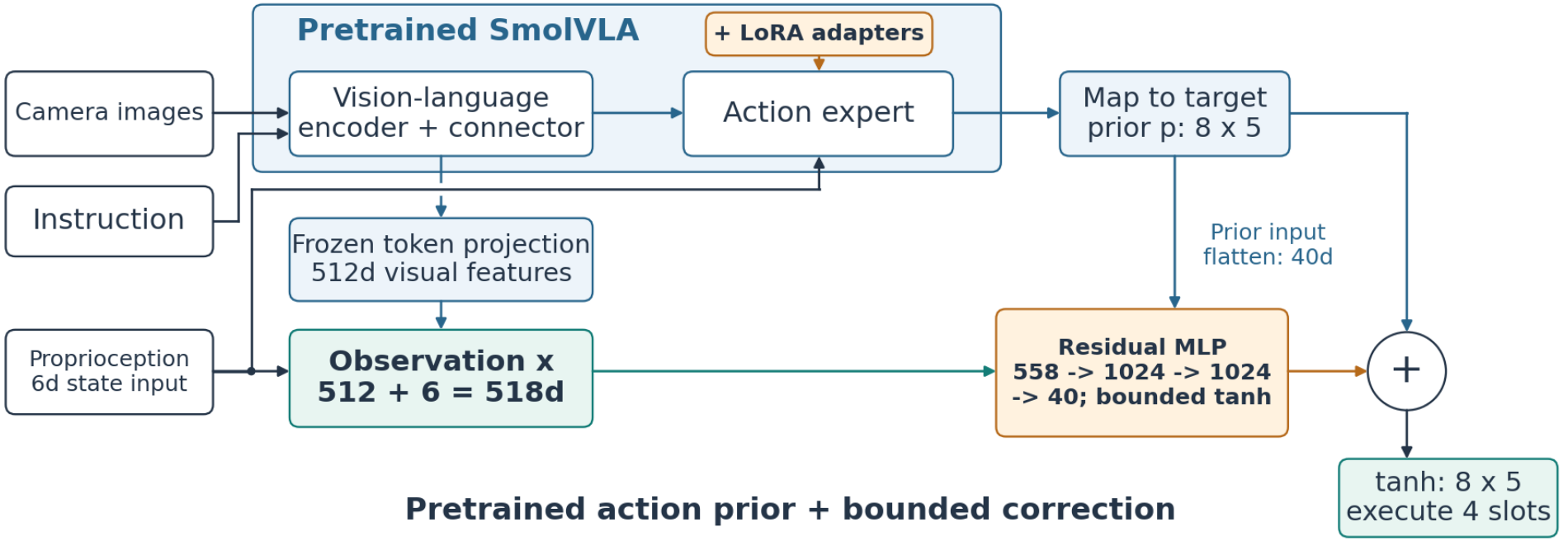
Empty gripper → Grasp + lift → Carry + place

Bootstrapping control on a new robot



Acquire → consolidate → compose. Pretrained SmolVLA + a newly initialized residual.

SmolVLA with a learned CDPR correction



Earlier phases: expert LoRA + residual | Full-task continuation: VLA / LoRA frozen; residual trained

CDPR: learn skills, retain them, then combine them

Learn with RL
Try a simple skill

→

Check retention
Test earlier skills

→

SIL if needed
Rehearse successes

→

Continue RL
Learn the next skill

Learn the parts

Approach, pickup and placement have separate instructions and starting states. Dense rewards guide initial progress; binary success rewards refine each skill.

Then learn the whole task

SIL means self-imitation learning. It retains earlier skills using successful actions. Before full-task RL, fit scripted-oracle and assisted complete-task trajectories. Then train from empty-gripper starts.

CDPR stage transitions were validation-guided and manual. Full-task RL and evaluation use unassisted starts.

RL: improve actions using rewards

Try the task → score the attempts → update the policy.

$$\mathcal{L}_{\mathrm{RL}} = \mathcal{L}_{\mathrm{clip}} - \eta\mathcal{H}$$

Small policy updates

PPO-style clipping reduces the incentive for large action-likelihood changes; it does not impose a hard bound on the update.

L_{RL} is the training loss.

L_{clip} is the clipped policy loss.

Compare attempts

Group-relative credit (GRPO-style) compares attempts from the same scene.

H encourages exploration.
η controls its strength.

Learning a new skill can erase an old one

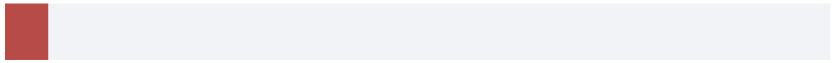
Reaching success

Before learning to carry and place



64.1%

After learning to carry and place



5.3%

What happened?

Placement practice starts with the object already held.

The policy learns transport and release, but its earlier reaching skill deteriorates.

Learning the latest skill can overwrite useful earlier behavior.

Isolated reaching test · three evaluation rounds per condition.

SIL: recover skills by imitating successful actions

SIL means **self-imitation learning**: fit the policy to its own successful actions.

$$\mathcal{L}_{\text{SIL}} = \text{mean}[(\text{predicted action} - \text{recorded successful action})^2]$$

Balance successful actions by instruction, fit the policy, check retention, then resume RL.

Separate recovery evaluation · reaching: 3 rounds; placement: 2 rounds per condition.

Skill	Before SIL	After SIL
Reaching	8.0%	31.1%
Plate placement	50.0%	52.3%
Bowl placement	23.05%	27.65%

Partial recovery; separate from the preceding forgetting test.

Full-task RL: keep credit for a successful pickup

After consolidation and oracle/assisted fitting: full-task RL · empty-gripper start

Approach ✓

Milestone achieved

→

Held pickup ✓

Milestone achieved

→

Placement ×

Task incomplete

Keep local credit

Approach and pickup retain their milestone-specific learning signal.

Train the remaining step

Suppress the negative completion bonus on achieved approach/pickup milestones.

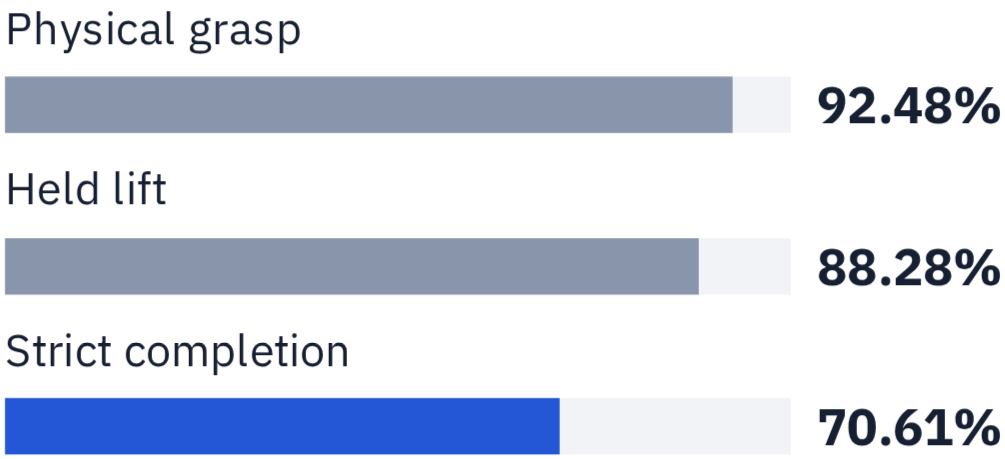
Preserve useful partial behavior while learning to finish.

Milestones assign training credit; evaluation uses no external skill switching.

CDPR: 70.61% strict completion in simulation

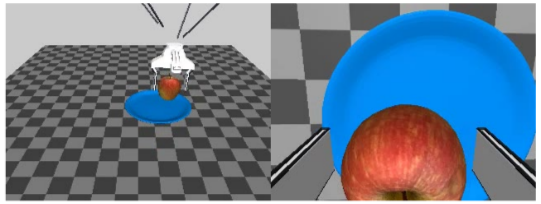
Development validation: 723 / 1,024 · 512 plate + 512 bowl attempts

Marginal outcome rates, all attempts

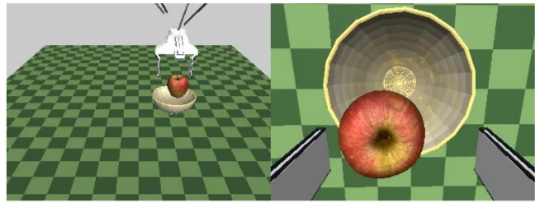


Zero-to-100% scale.
One training lineage.
Development validation.

Selected historical successes



Apple into plate



Apple into bowl

Historical examples; separate from the 70.61% endpoint.

Transport and release remain the main bottleneck.

Strict success: grasp, held lift, intentional correct release; no carry slip or wrong placement. All attempts include numerical failures.

Full-task completion across destination types

Outcome	All attempts
Strict completion	723/1,024 · 70.61%
Plate completion	393/512 · 76.76%
Bowl completion	330/512 · 64.45%
Carry slip	16.70%
Wrong placement	14.16%

Development validation used during training. Empty-gripper starts, one instruction, no teacher assistance. All attempts retained; event flags can overlap.

CDPR adaptation: procedure and observed outcome

Three problems

New action interface

Too few successful complete rollouts

Earlier skills degrade during new training

Three responses

SmolVLA prior + learned residual

Simple skills first; binary milestones later

Balanced self-imitation + oracle/assisted fitting

The complete pipeline produced a policy that grasps, carries and places under one instruction.

Partial skill recovery and 70.61% development-validation completion. Each ingredient’s independent contribution remains to be tested.

Part 3 · What comes next

01
Planning above a fixed controller

02
Learning for a new embodiment

03
What comes next

What should the robot predict—and when should it ask for help?

Physical understanding and decisions under uncertainty

Planned VLA work

01 **Predict physical consequences**

PINN / iPINN and JEPA: test contact and task-outcome predictions.

02 **Estimate when the model may be wrong**

Calibrate error estimates to trigger replanning, verification or human help.

03 **Choose useful experience**

Test selective simulation and replay at matched compute.
Engineering-control cartography motivates a VLA extension.

Near term: independent evaluation and physical deployment tests.

What the experiments establish

Planning changes both successes and failures.
Count rescued and lost successes alongside the executor’s own variability.

Adaptation requires learning and retaining skills.
The combined CDPR pipeline reached 70.61% strict success in simulation development validation.

The next checks are independent evaluation and hardware tests.
Use matched component comparisons to establish which training stages contribute.

References: policies, planning and reinforcement learning

Kim, Finn & Liang (2025). Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success.

Li et al. (2025). SimpleVLA-RL: Scaling VLA Training via Reinforcement Learning.

Shukor et al. (2025). SmolVLA.

Xiao et al. (2025). Probe, Learn, Distill.

Bootstrapping Vision-Language-Action Models for New Robot Embodiments with Staged Reinforcement Learning — manuscript draft

Study manuscripts: planner evaluation and staged VLA adaptation.

References: related workshop publications

Salloum, Mifsud Bonici, Ibrahim, Osinenko & Kornaev (2026). Diversity-Aware Adaptive Collocation for Physics-Informed Neural Networks via Sparse QUBO Optimization and Hybrid Coresets.
AI&PDE, ICLR 2026 Workshop.

Degtiarev, Ryabkova & Kornaev (2026). LLM-Driven Loss Balancing for Physics-Informed Neural Networks.
AI&PDE, ICLR 2026 Workshop.

Nurtdinov, Korshuk, Kornaev & Maloletov (2025). Evaluation of a Robust Control System in Real-World Cable-Driven Parallel Robots.
ICLR 2025 Workshop: Towards Agentic AI for Science.

Related publications; workshop venues identified explicitly.