Penn Engineering
University of Pennsylvania

GMLR

CIS 6280 · FALL 2026

WORLD MODELS

LECTURE 18

World Models for Robotics II
Vision-Language-Action and World-Action Models

Jiatao Gu

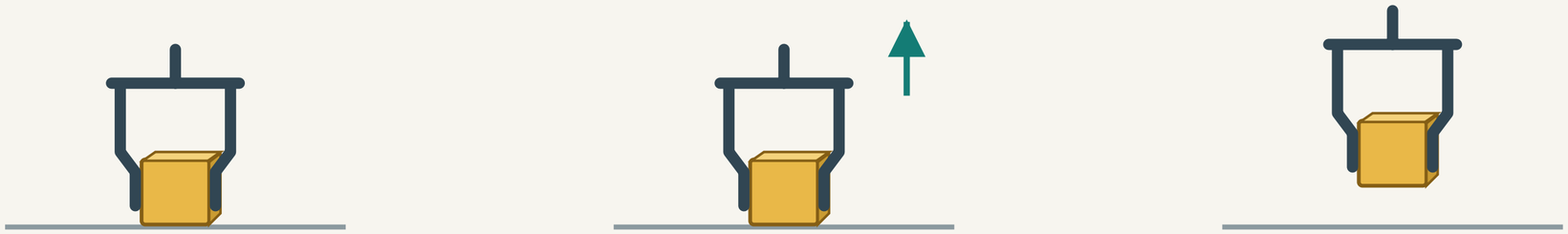
University of Pennsylvania · November 17, 2026

Announcements

Wed, Nov 18	Assignment 3 is due.
Thu, Nov 19	LLM as World Models.
Final project	Cloud compute budget: $100 per team. The final report and code are due December 14.

One robot step in a grasp demonstration

Instruction g_t “Lift the yellow cube.”



o_t · Observation

The fingers hold the cube at table height.

a_t · Action

Raise the wrist slightly and keep the fingers closed.

o_{t+1} · Next observation

The next camera frame shows the cube off the table.

A vision-language-action (VLA) policy chooses the commands

$$\pi_\theta(a_t \mid o_t, g_t)$$

Read the bar as “given.” Which action fits this observation and instruction?

What the robot knows	The gripper is holding the cube at table height. The instruction is “lift the yellow cube.”
What the policy predicts	One wrist-and-gripper command, for example “raise the wrist with fingers closed.” The demonstrator’s command log supplies the training target.

OpenVLA · project and robot videos · Physical Intelligence · π_0

RT-2

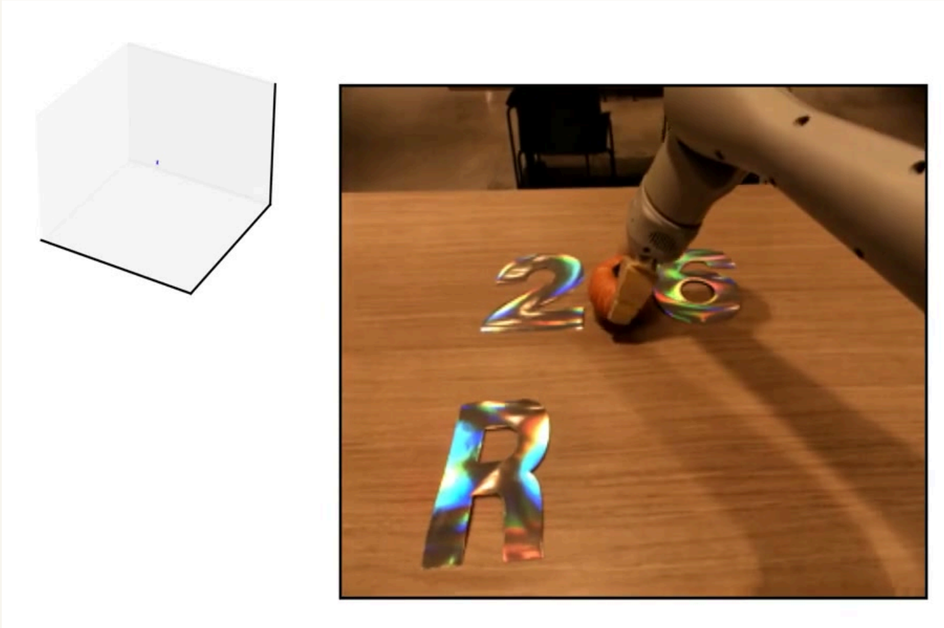
Google DeepMind · 2023

VLA · Action tokens from a vision-language model

$$\pi_\theta(a_t \mid o_t, g_t)$$

Input	A camera image and a language instruction.
Output	Tokens encoding one wrist-and-gripper command.
How it works	Co-finetune a vision-language model on robot demonstrations and web vision-language tasks. Express commands in its token vocabulary.

RT-2 turns an instruction into a robot command



DeepMind · RT-2 · “move pumpkin to 2” · physical robot

What does “2” refer to?

The model sees the marked table and reads the instruction. Its vision-language training helps it interpret the numeral.

What leaves the model?

A sequence of tokens encodes a gripper command. Robot demonstrations teach the model which tokens move the arm.

OpenVLA

Stanford, UC Berkeley, Toyota Research Institute and collaborators · 2024

VLA · Open model for generalist manipulation

$$\pi_\theta(a_t \mid o_t, g_t)$$

Input	The current robot camera image and an instruction.
Output	Discrete tokens for end-effector movement and gripper control.
How it works	Train a 7B vision-language backbone on 970,000 robot episodes from Open X-Embodiment, then adapt it to the target robot.

OpenVLA · project and robot videos

OpenVLA learns action tokens from a robot demonstration



OpenVLA · WidowX rollout · “Put corn on plate” · source clip at 1.5×

Inputs $o_t,\ g_t$

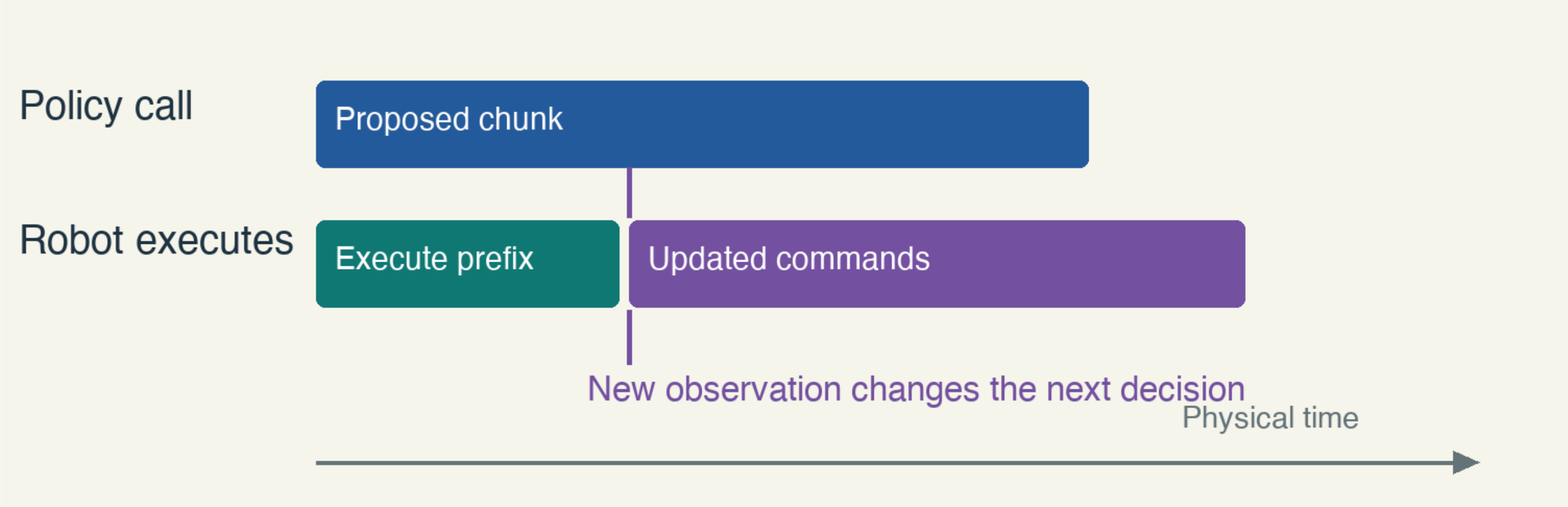
The model receives the current camera view and the instruction.

Target a_t

The demonstrator’s wrist motion and gripper command at this instant are encoded as tokens. Decode them to move the robot.

OpenVLA · project and robot videos

An action chunk spans several robot time steps



A proposed chunk extends beyond the prefix the robot executes before seeing the scene again.

$$A = a_{t:t+H-1}, \qquad o_{t+1:t+H} \text{ are the observations over those } H \text{ steps}$$

The policy proposes a chunk of H commands. It can revise the remaining commands after a fresh observation.

π_0

Physical Intelligence · 2024

VLA · Flow matching over continuous action chunks

$$\pi_\theta(A \mid o_t, g_t), \qquad A = a_{t:t+H-1}$$

Input	Camera images, robot state and a language instruction.
Output	A chunk of H continuous robot commands.
How it works	A pretrained vision-language backbone encodes context. An action expert learns a flow field that transforms noise into a command sequence.

How π_0 generates a continuous action chunk



Physical Intelligence · π_0 folding laundry · Figure 2

Vision and language	The VLM encodes the camera images and instruction.
Robot state and actions	The action expert takes the robot state and a noisy action chunk. It attends to the visual and language context.
Execution	Starting from noise, the expert refines the chunk into continuous commands for the robot.

Two clocks in a flow policy

$$A_\tau = (1-\tau)\,\epsilon + \tau A, \qquad v_\theta(A_\tau, \tau \mid o_t, g_t) \text{ is trained toward } A - \epsilon$$

Sampling time τ

A is a recorded action chunk and ε is noise of the same shape. The policy refines the whole chunk while the current observation and instruction stay fixed.

Robot time

After sampling, the controller executes commands from the chunk in order and obtains a new observation.

$\pi_{0.5}$

Physical Intelligence · 2025

VLA · Subtask prediction and continuous control

$$p_\theta(\text{subtask} \mid o_t, g_t), \qquad \pi_\theta(A \mid o_t, \text{subtask})$$

Input	The current scene, robot state and a household task instruction.
Output	A subtask such as “pick up the plate,” followed by an action chunk.
How it works	Co-train on heterogeneous robot, semantic and web data to carry out longer tasks in previously unseen homes.

Physical Intelligence · $\pi_{0.5}$ · Fig. 7a

$\pi_{0.5}$ chooses the next subtask in a new home

Home 1

Human: “put the items in the drawer”

HL prediction:


pull out the drawer


pull out the top right drawer


pick up tong

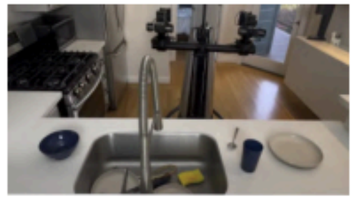

put tong into drawer


push the top drawer

Home 2

Human: “place the dishes in the sink”

HL prediction:


pick up plate


put plate in the sink


put cup in the sink


pick up the spoon


pick up bowl

Physical Intelligence · Fig. 7a, first two homes · each blue caption is a model-predicted subtask.

“Place the dishes in the sink” stays fixed. The robot chooses a new subtask as the scene changes.

$\pi_{0.7}$

Physical Intelligence · 2026

VLA · A policy guided by language and goal images

$$\pi_\theta(A \mid o_t, g_t, \text{optional goal image})$$

Input	Robot observations and a language instruction or an image of the desired result.
Output	Continuous commands that carry out the supplied instruction.
How it works	Train with richer task context so people or a higher-level system can steer the same action policy through new tasks.

$\pi_{0.7}$ can be coached one step at a time



Physical Intelligence · Fig. 14 · human coaching during an air-fryer task.

Who supplies the next goal?

Here a person tells the robot to open the fryer, put the sweet potato inside, then close it. $\pi_{0.7}$ turns each instruction into movement.

Another way to prompt the policy

The policy can also take a subgoal image showing the desired scene. A separate model can generate that image.

GR00T N1.5

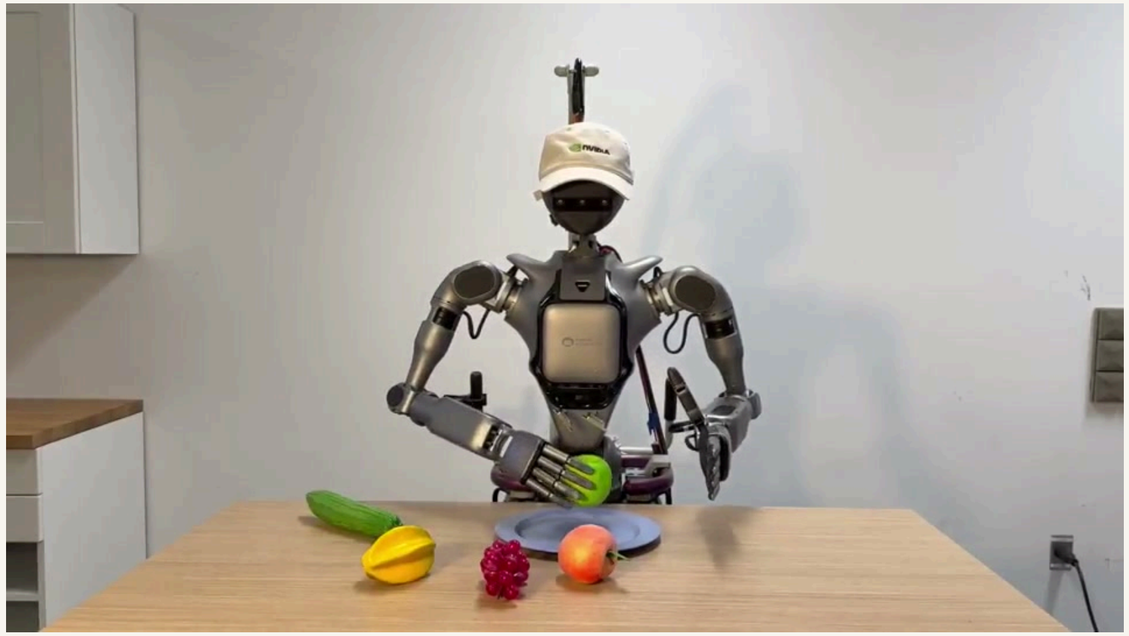
NVIDIA Research · 2025

VLA · Humanoid policy with future-feature supervision

$$\pi_\theta(A \mid o_t, g_t), \qquad \mathcal{L} = \mathcal{L}_{\text{action}} + \lambda \mathcal{L}_{\text{future features}}$$

Input	Robot camera views, language and the body configuration.
Output	An action chunk for the arm and hand.
How it works	Add FLARE, an objective that matches features of a later camera frame, alongside the flow-matching action loss.

GR00T N1.5 learns from where the apple ends up



NVIDIA · GR00T N1.5 · “Pick the apple from table to plate”

During training

An encoder describes a later camera frame in features. An extra objective teaches the policy to match those future features.

On the robot

The instruction, current view and robot state guide a flow-based action head. Its output drives the arm and hand.

A forward world model predicts the effect of a command

$$p_\omega(o_{t+1} \mid o_t, a_t)$$

Lift with fingers closed	From this grasp, the model can predict that the cube rises with the hand.
Lift with fingers opening	Keep the starting view fixed. Supply a different command. The model can predict that the hand rises alone.

PointWorld · physical experiments · DreamDojo · project

A world-action model (WAM) learns actions and their outcomes

$$p_\theta(a_t, o_{t+1} \mid o_t, g_t)$$

One demonstration, two targets

The robot log records the lift command. The next camera frame shows whether the cube came along.

The VLA policy inside it

$\pi_{\mathrm{WAM}}(a_t \mid o_t, g_t) = \int p_\theta(a_t, o_{t+1} \mid o_t, g_t)\, do_{t+1}$

Averaging over possible outcomes leaves a distribution over commands.

DreamZero · paper · Fast-WAM · Fig. 1 and real-world latency comparison

Why generate future video for a robot?

Specify a visible outcome	Show the object being grasped, moved and placed. A policy can use those future views as a visual plan.
Learn what actions change	Robot recordings pair commands with the resulting images. Video prediction gives dense supervision about the interaction.
Practice and evaluate	An action-conditioned video model can supply observations for a policy rollout or create new training demonstrations.

VERA · MIT CSAIL · DreamZero · paper · DreamDojo · project

VERA

MIT-led collaboration with CMU, Amazon FAR and independent researchers · 2026

Robot policy · Video planning followed by inverse dynamics

$$\underbrace{p_\theta(o_{t+1:t+H} \mid o_t, g_t)}_{\text{visual plan}} \qquad \underbrace{p_\phi(A \mid o_t, o_{t+1:t+H})}_{\text{inverse dynamics}}$$

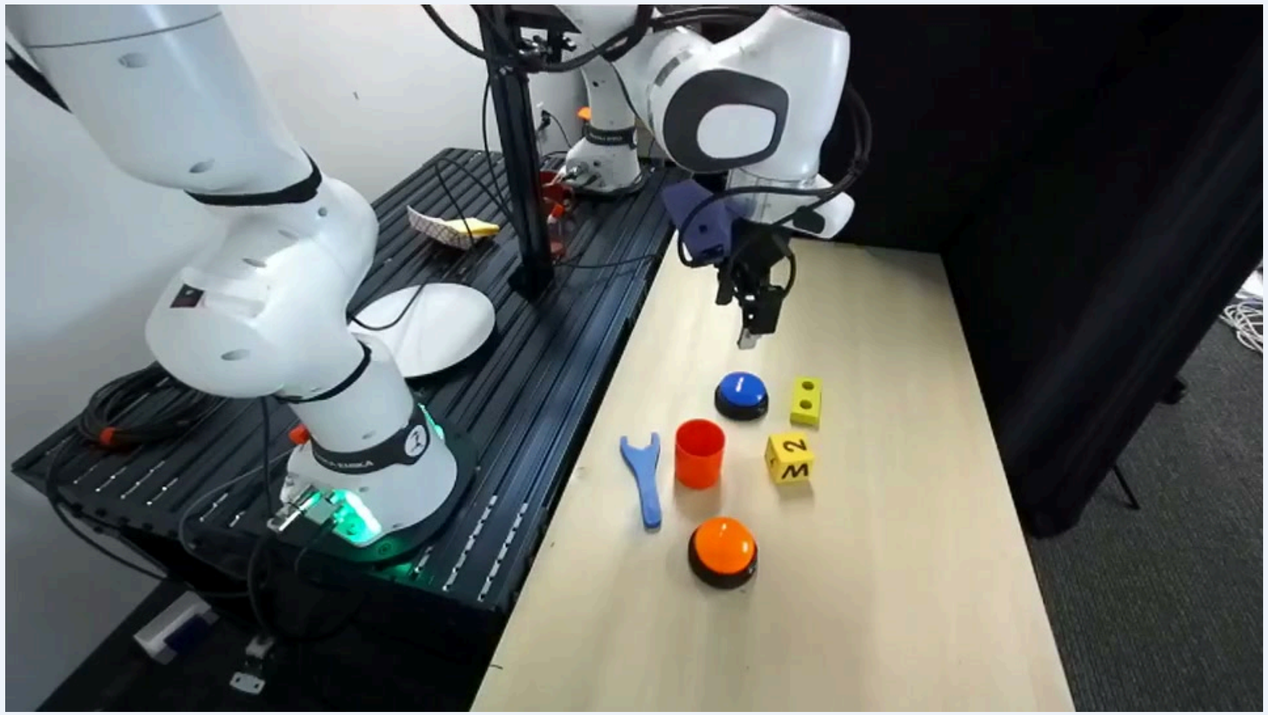
Input	Recent camera views, the task instruction and the robot interface.
Output	A visual trajectory translated into executable robot commands.
How it works	Keep the video planner fixed. Learn a robot-specific inverse model relating small commands to pixel motion through a Jacobian.

How could a robot act from a predicted future?

$$p(a_t, o_{t+1} \mid o_t, g_t) = \underbrace{p(o_{t+1} \mid o_t, g_t)}_{\text{predict the next view}} \; \underbrace{p(a_t \mid o_t, g_t, o_{t+1})}_{\text{infer the command}}$$

Predict a visual plan	Given “stack the cube,” generate a video in which the cube reaches the block.
Find the commands	Given the current view and that proposed motion, infer how this gripper would have to move.

Can a robot follow a video it imagined?



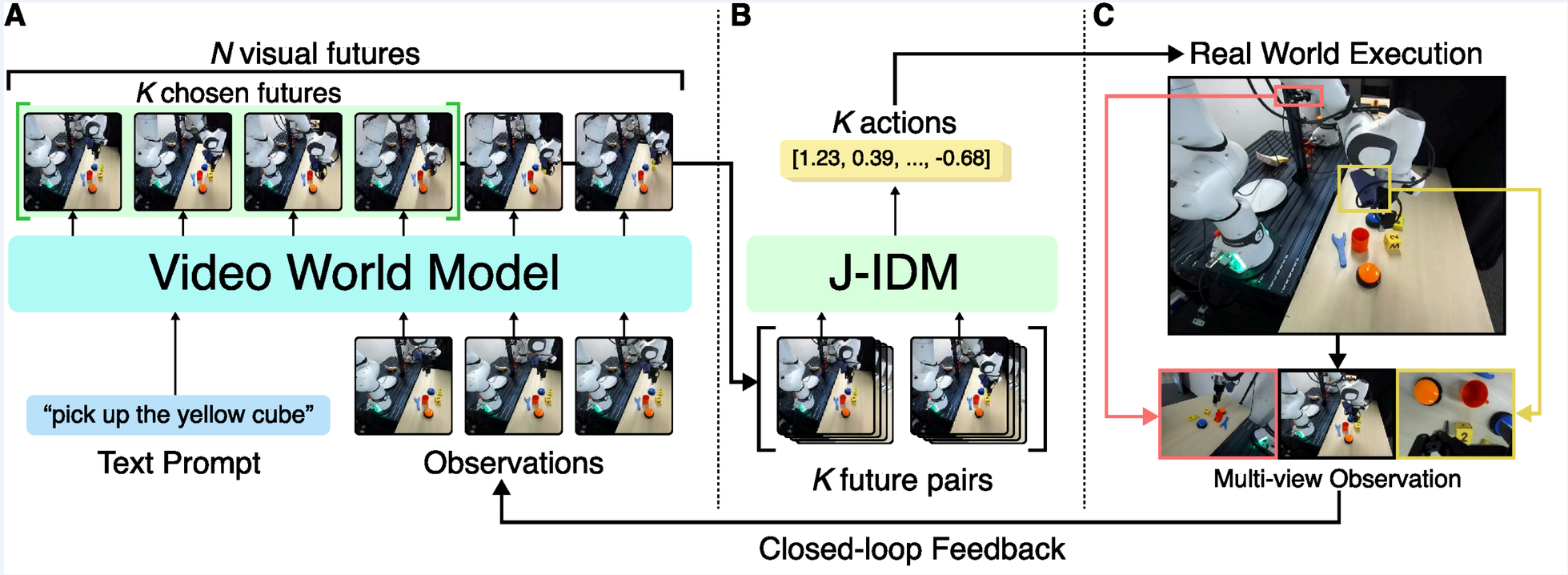
VERA · real Panda arm · “place the yellow cube on top of the lego block”

This Panda takes end-effector pose increments and a gripper command. Which commands could produce the motion in a visual plan?

Which part of a recorded transition is missing?

A robot log	Camera view before the move · command sent · camera view after the move.
Cover the last view	Given the starting view and command, forward dynamics predicts the result.
Cover the command	Given the starting view and a desired next view, inverse dynamics proposes a command that could connect them.

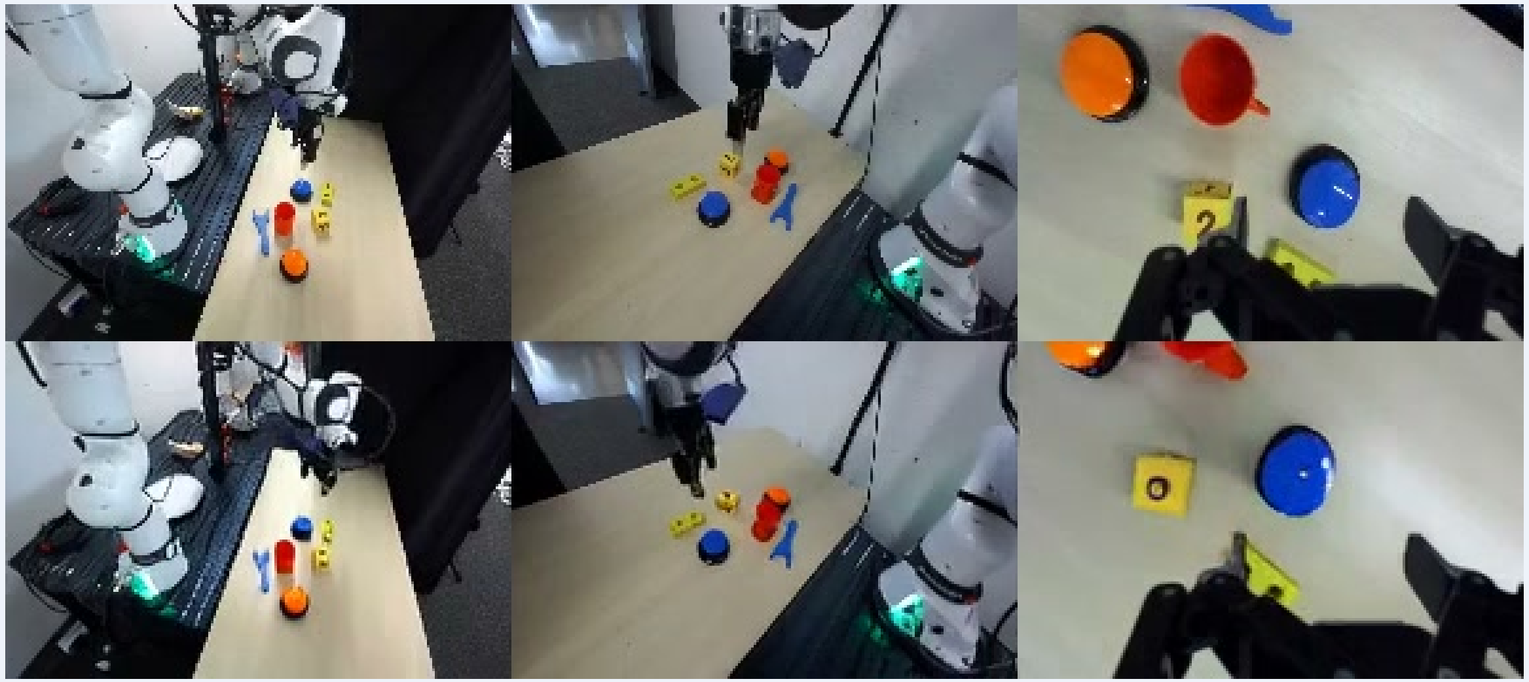
How VERA turns a desired image change into a command



VERA · a visual plan supplies desired motion, the inverse model infers commands, and the cameras observe the result.

The learned Jacobian relates a small robot action to local pixel motion. The translator uses the desired pixel motion to infer an action.

When can you tell whether the cube was grasped?



VERA · top row: three real cameras · bottom row: three predicted views of the same cube-stacking task

Follow the yellow cube as the gripper closes and lifts. Which camera view gives you evidence that the cube moved with it?

Which camera reveals the hidden blue button?



VERA · camera views of the tabletop, the wall and an orange distractor

Instruction

Push the blue button.

Inspect the views

Locate the blue button. What could the robot infer incorrectly if it saw only the orange distractor?

DreamZero

NVIDIA · 2026

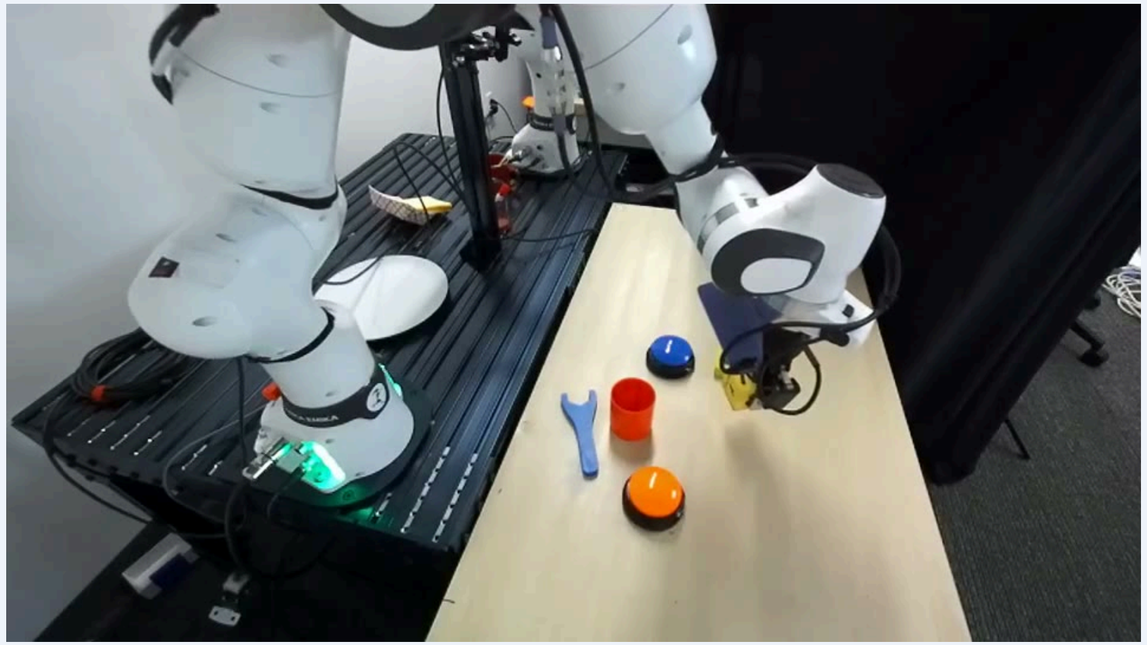
WAM · A VLA policy that jointly predicts video and actions

$$p_\theta(a_{t:t+H-1}, o_{t+1:t+H} \mid o_t, g_t)$$

Input	Recent camera frames, robot state and the task instruction.
Output	A command chunk and future camera views covering the same interval.
How it works	Start from a pretrained video diffusion model. Jointly learn video and action denoising on robot recordings.

DreamZero · project · DreamZero · paper

What did the camera record beyond “close the gripper”?



Cube-stacking example from VERA · the camera records the interaction.

The command log

It records the command to close and lift.

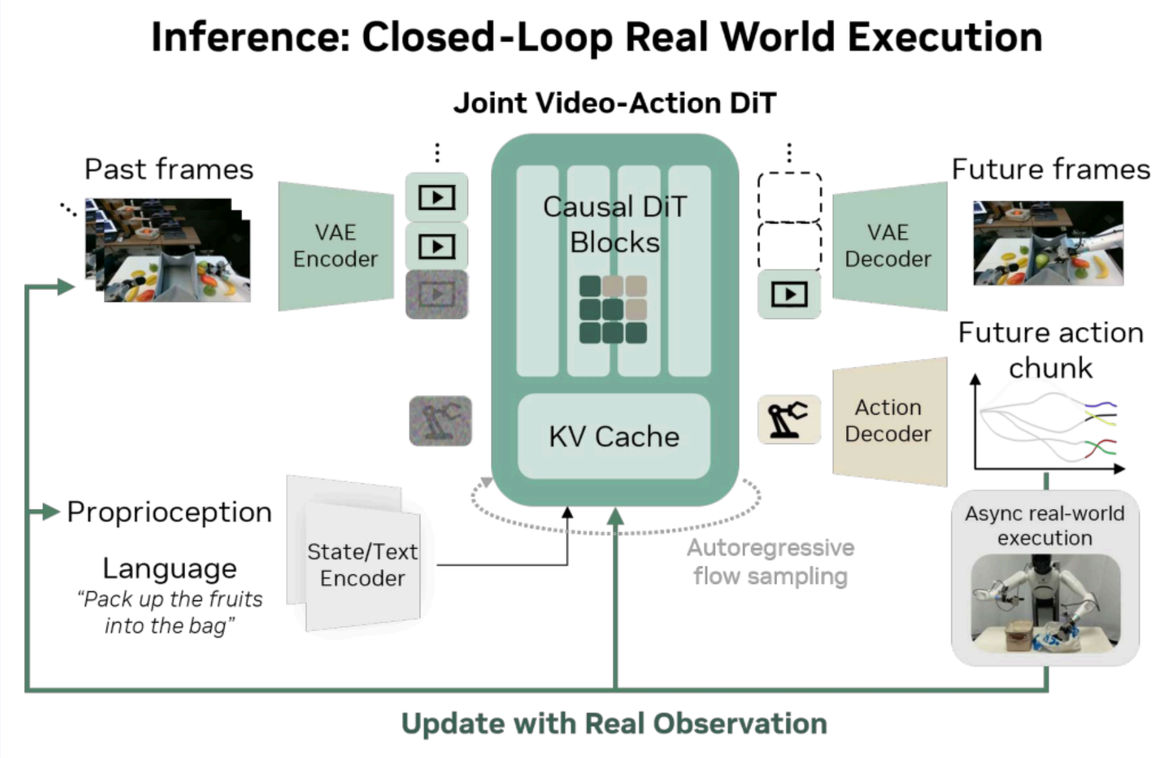
The next views

They show whether the cube rises with the hand or stays on the table.

The joint target $(A, o_{t+1:t+H})$

DreamZero uses commands and the views from the same interval as paired training targets.

One DreamZero control cycle



DreamZero · Figure 4, inference panel · the green path carries fresh observations.

Condition on the real scene

Recent camera frames, the instruction and robot state condition the shared video–action transformer.

Generate the next chunk

During training the model jointly denoises recorded video and action chunks with noise added. At rollout it samples future views and commands.

Execute and observe

Send the commands to the robot. Its cameras supply the real frames used as history for the next chunk.

What the physical rollout adds



DreamZero · fruit packing on AgiBot · excerpt from the authors' 2× playback

Watch the object as well as the arms. Mark the first moment an execution error would require a new observation.

Dream-1

Dream Labs · October 2026

WAM policy · Joint video–action prediction across robot bodies

$$p_\theta(A, o_{t+1:t+H} \mid o_{\le t}, g_t)$$

Input	Head and wrist-camera observations, plus a task instruction.
Output	Future views and action chunks; gripper-pose outputs support transfer.
How it works	Predict video and actions jointly. Each target robot converts gripper poses into joint commands with its own IK.

The same gripper command can drive a different robot



Dream Labs · Dream-1 on a real Metal Arm · banana into bowl · author playback at 4×

Predict the gripper target

Specify the desired position, orientation and gripper opening.

Use the target robot's IK

Inverse kinematics converts that target into joint commands.

Check feasibility

Reachability, arm paths and IK quality still depend on the robot.

Unified World Models

University of Washington and Toyota Research Institute · RSS 2025

UWM · One model for policies, forward and inverse dynamics

$$p_\theta(a_t, o_{t+1} \mid o_t)$$

Input	The current observation, plus any action or future view supplied by the query.
Output	The missing action, future view, or both, depending on the query.
How it works	Use independent diffusion times for actions and video. Fix known inputs, denoise requested outputs and leave hidden inputs fully noisy.

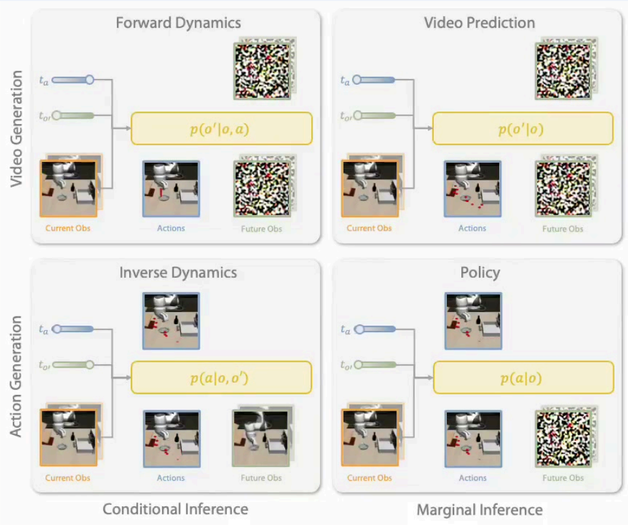
Forward WM, WAM and UWM on the same grasp

Model	What do we give it?	What do we get back?
Forward world model	A candidate command: “Open the gripper before lifting.”	A predicted outcome: the hand lifts and the cube stays behind.
WAM policy · DreamZero	A task: “Stack the cube.” The policy chooses the commands.	A command sequence plus the future views it expects to accompany it.
Unified World Model (UWM)	The same scene. We can supply a command, supply a desired future view, or supply neither.	Given a command, predict the view. Given the view, infer commands. With only the scene, it can act as a policy.

To test a new command, hold the starting scene fixed and change the action supplied to the model.

How one UWM answers four questions

$$p_\theta(a_t, o_{t+1} \mid o_t)$$



Zhu et al. · four inference modes of the same trained UWM

UWM query	Prediction	What we supply
Forward dynamics	$p_\theta(o_{t+1} \mid o_t, a_t)$	Command
Inverse dynamics	$p_\theta(a_t \mid o_t, o_{t+1})$	Desired future view
Policy	$p_\theta(a_t \mid o_t)$	Current observation
Video prediction	$p_\theta(o_{t+1} \mid o_t)$	Current observation

Every row starts from the current observation and uses the same weights. Keep a supplied stream clean, denoise the one to be predicted, and leave a hidden stream at full noise.

How does UWM learn from a video with no commands?



UWM · physical rollout from the Rice-Cooker task

Robot recording with commands

Train on the recorded action and the following images. The two streams can receive different amounts of noise.

Recording with images alone

Hide the action stream with noise and train the future-image prediction. The visual sequence still supplies a target.

DreamDojo

NVIDIA-led collaboration · 2026

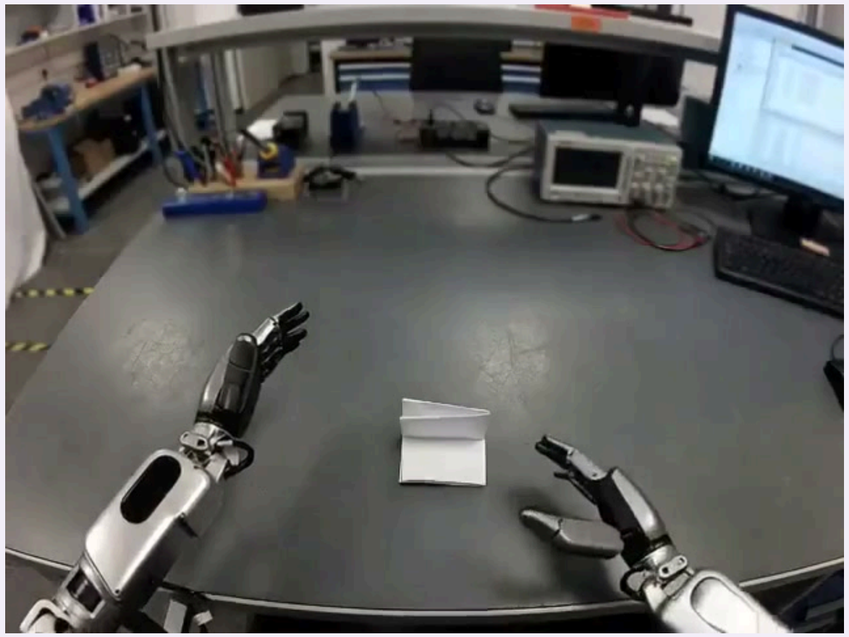
Forward world model · Action-conditioned video simulator

$$p_\theta\left(o_{t+1:t+H} \mid o_{\le t}, a_{t:t+H-1}\right)$$

Input	Camera history and commands supplied by an external robot policy.
Output	Predicted camera views after those commands.
How it works	Pretrain on large-scale human video with inferred latent actions, then adapt to continuous commands from a target robot.

DreamDojo · project · DreamDojo · Figure 5a

Who supplies the next camera view in a simulator?



DreamDojo · generated GR-1 robot-camera rollout

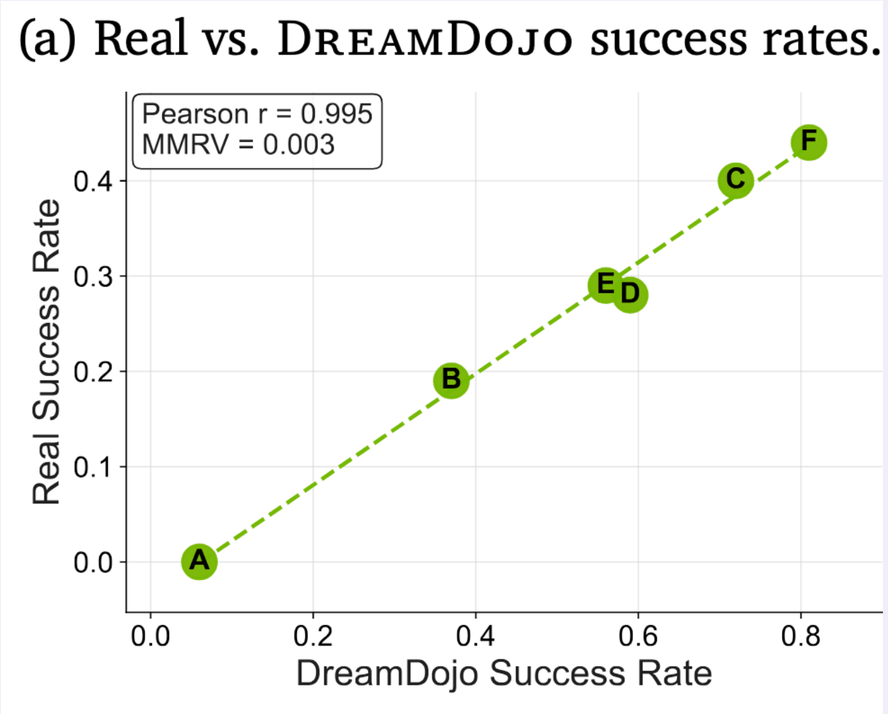
The policy chooses

An existing robot policy receives a view and sends a command to DreamDojo.

The simulator answers

DreamDojo generates the next view under that command. The policy uses this generated observation to choose again.

Would this simulator help you choose a checkpoint?



DreamDojo · six checkpoints of the same GR00T N1.5 policy on AgiBot fruit packing.

Read a point

Horizontal: simulated packing score. Vertical: hardware score. Credit depends on how many fruits reach the bag.

Choose a policy

If you could deploy one checkpoint, which would you choose?

Estimate its hardware score

Read its vertical coordinate on the hardware axis. How close is it to the simulated score?

DreamDojo · Figure 5a

DreamGen

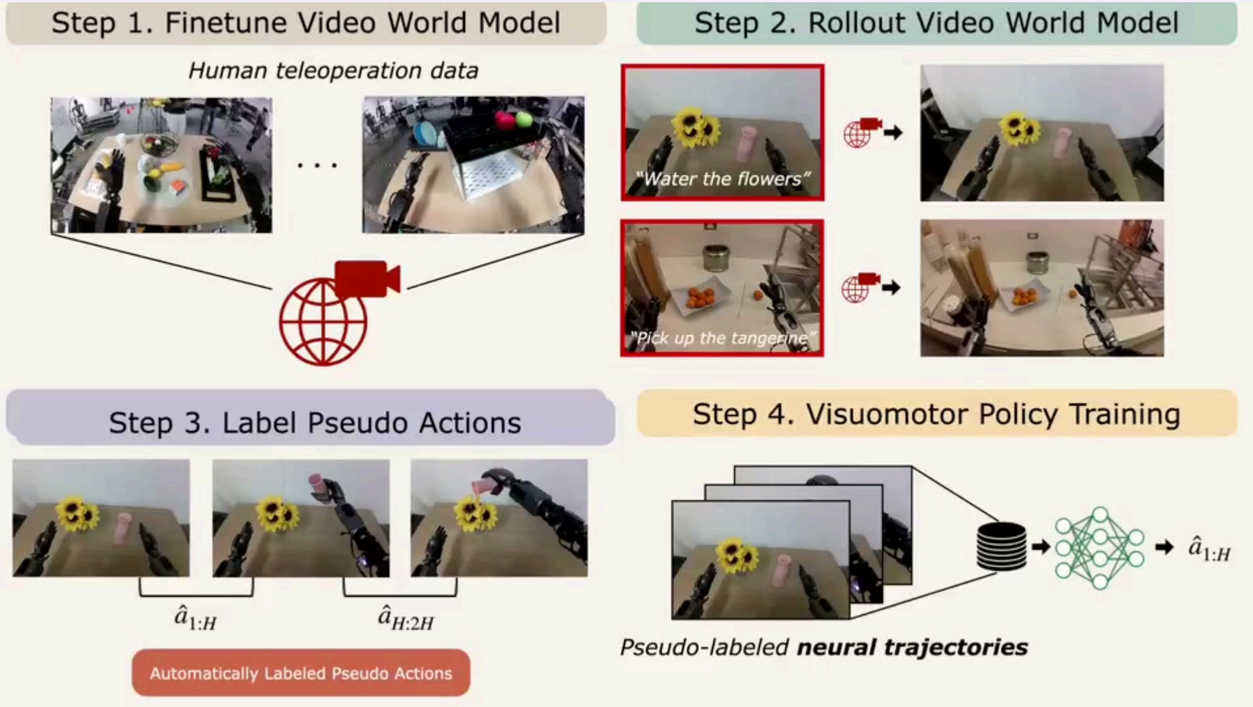
NVIDIA Research · 2025

Training-data pipeline · Video generation and inferred actions

$$\underbrace{p_\theta\left(o_{t+1:t+H} \mid o_t, g_t\right)}_{\text{synthetic demonstration}} \qquad \underbrace{p_\phi\left(A \mid o_{t:t+H}\right)}_{\text{action labels}}$$

Input	An initial scene, a desired task and a video model adapted to the robot.
Output	Synthetic observation-action demonstrations for policy training.
How it works	Generate robot videos, infer missing actions with an inverse or latent-action model, and train a policy on the resulting trajectories.

A world model can also make training data



DreamGen · author animation of data generation and policy training.

Generate a visual demonstration, infer the missing action labels, then train a robot policy on those pairs.

PointWorld

Stanford University and NVIDIA · 2026

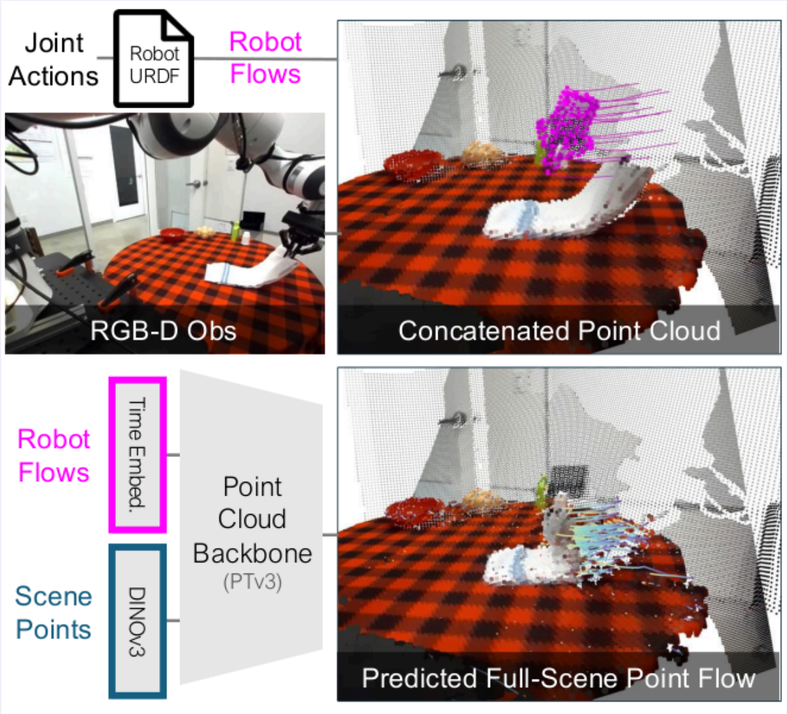
Forward world model · Action-conditioned 3D point motion

$$\widehat{\Delta P}_{t+1:t+H} = f_\theta(P_t, a_{t:t+H-1})$$

Input	Scene points P_t from RGB-D views and a candidate robot action sequence.
Output	3D displacements ΔP of scene points over the prediction interval.
How it works	Represent robot motion and scene response as 3D point flows, allowing different robot bodies to share a geometric description.

PointWorld · physical experiments · Huang et al. · PointWorld · Fig. 2 and planning protocol

PointWorld predicts how the scene moves under a push



PointWorld · Fig. 2 · magenta marks where the robot would move.

Start with visible surfaces

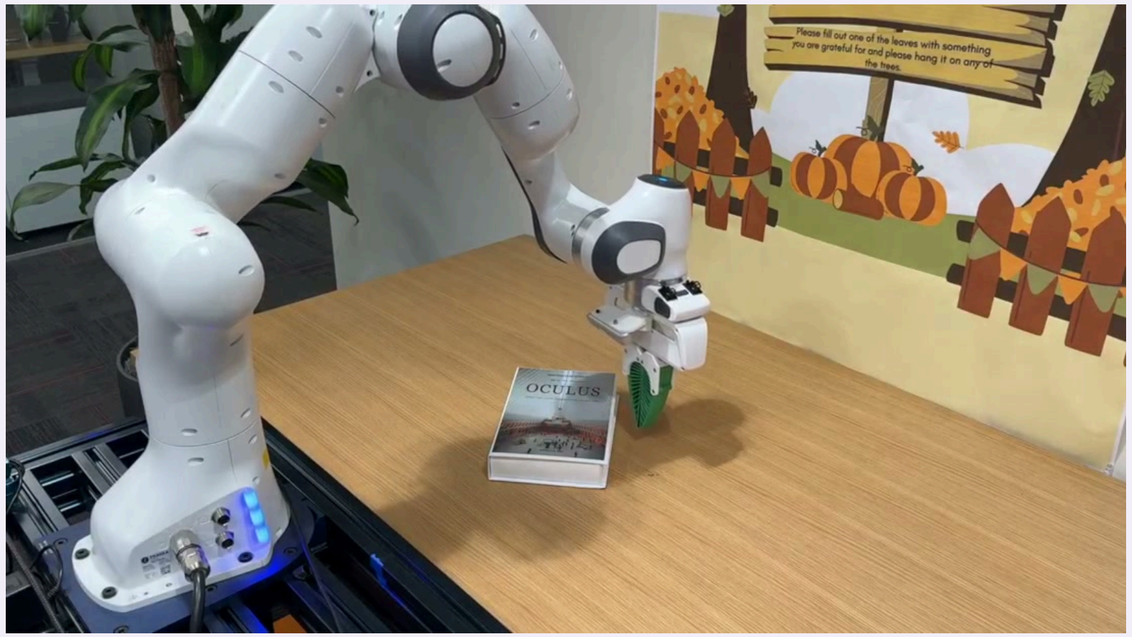
A depth camera gives a 3D position for each visible scene point. The proposed joint motion gives a path for points on the gripper.

Predict the effect

For that gripper path, predict where the scene points will move. Try a different push on the same observation and the forecast can change.

Huang et al. · PointWorld · Fig. 2 and planning protocol · PointWorld · physical experiments

Which push puts the book at its target?



PointWorld · author footage of a Franka pushing a book on a real table.

Specify the goal

A person marks the object and its target position. The planner samples possible arm motions.

Score the pushes before acting

For each motion, PointWorld predicts the book's movement. Favor a push that brings its points close to the target. The arm executes the selected plan without replanning in this experiment.

PointWAM

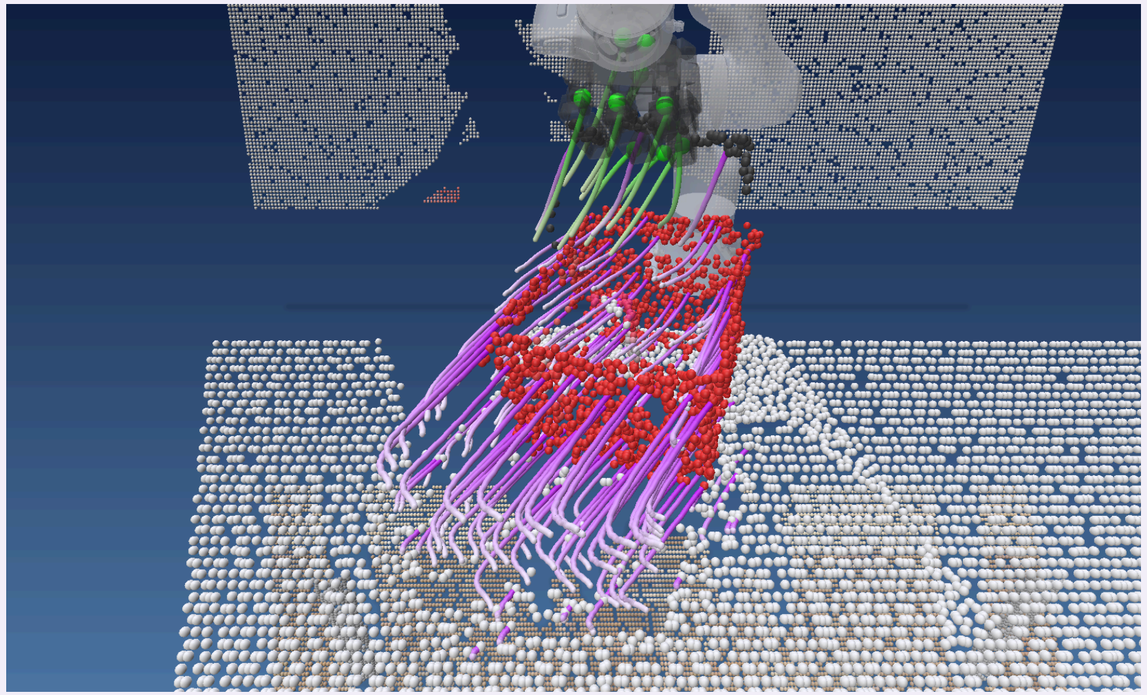
POSTECH, KAIST, RLWRLD and Hanyang University · 2026

3D WAM · Joint hand and scene trajectory prediction

$$p_\theta(P^{\text{hand}}_{t+1:t+H}, P^{\text{scene}}_{t+1:t+H} \mid o_t, g_t)$$

Input	The scene, current hand keypoints and a task instruction.
Output	Hand and scene point trajectories. A decoder converts hand paths to robot commands.
How it works	Pretrain on human hand-and-scene trajectories, then learn to translate predicted hand motion into actions for the target robot.

PointWAM predicts where the hand should go, too



PointWAM · Fig. 3 · pick-bucket task in DexJoCo simulation. Green tracks the hand. Purple tracks the scene.

A different input

Give the model the scene, hand keypoints and a task instruction. It predicts future hand and scene trajectories together.

From paths to motors

A decoder trained on robot demonstrations takes the predicted hand paths and the current robot state and outputs a chunk of commands.

DINO-WM

New York University and Meta FAIR · 2024

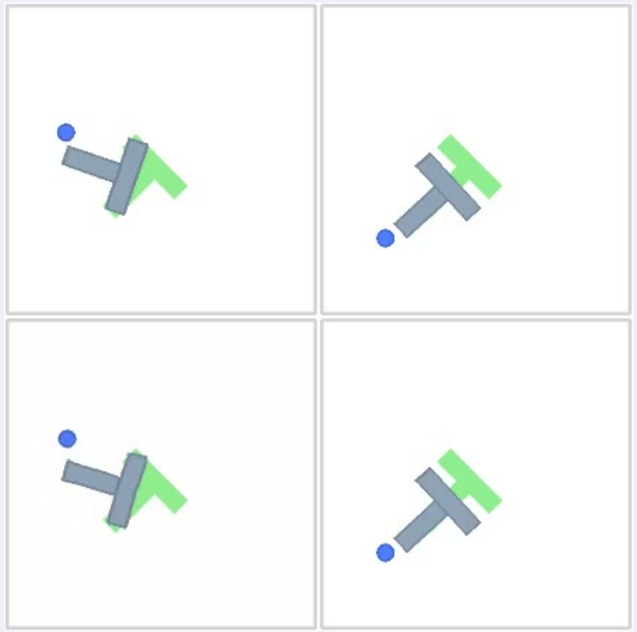
Forward world model · Predict visual features for planning

$$z_t = E(o_t), \qquad \hat{z}_{t+1} = f_\theta(z_{\leq t}, a_t)$$

Input	Visual patch features z from a pretrained encoder E, plus a candidate action.
Output	Predicted features of the next observation.
How it works	Learn dynamics from recorded trajectories. Search for actions whose predicted features approach the features of a goal image.

Zhou et al. · DINO-WM · PushT planning videos

DINO-WM scores pushes in a learned feature space



DINO-WM · simulated PushT · left: executed push (top), visualized forecast (bottom). Right: goal.

Learn what a push changes

Encode recorded frames as grids of DINOv2 patch features. From earlier features and actions, train a predictor to match the features of a later frame.

Use a goal picture

Encode the goal with the same encoder. Try several pushes in the model and choose one whose predicted feature grid comes close to the goal grid.

V-JEPA 2-AC

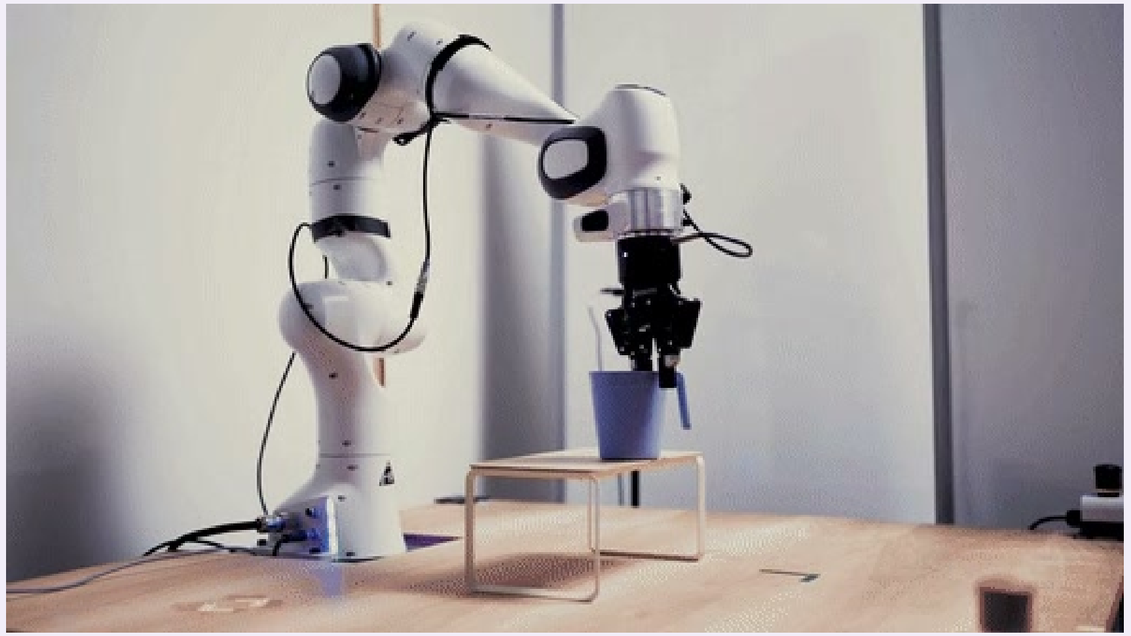
Meta FAIR · 2025

Forward world model · Action-conditioned latent prediction

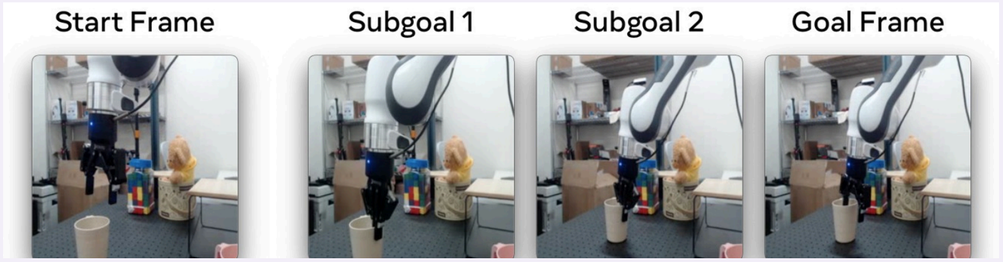
$$z_t = E(o_t), \qquad \hat{z}_{t+1} = f_\theta(z_{\leq t}, a_t)$$

Input	Encoded camera history and candidate robot commands.
Output	Predicted visual features for comparison with a goal image.
How it works	Start with video self-supervised learning. Add action-conditioned prediction using robot trajectories, then plan with the learned model.

V-JEPA 2-AC replans from the next real camera frame



Meta FAIR · physical Franka arm · excerpt from the authors' demonstration.



V-JEPA 2 · Fig. 14c, one cup trial · the two intermediate goal images and final goal are supplied.

Score candidate commands by distance to the current goal in feature space. Execute the first command, observe the robot again, and replan.

What do we compare when making a robot decision?

What the model predicts	Decision	What is compared?
Future camera view · DreamDojo	Which policy should we deploy?	Success rates of different policies inside the simulator.
3D scene points · PointWorld	Which way should we push the book?	Predicted positions of the book’s points versus their target positions.
Visual features · DINO-WM / V-JEPA 2-AC	Which command brings us nearer the goal?	Predicted visual features versus the features of the goal photo.

If the gripper slips, which signal would let the robot notice in time to change its next move?

DreamDojo · Figure 5a · Huang et al. · PointWorld · Fig. 2 and planning protocol · Zhou et al. · DINO-WM · PushT planning videos

Fast-WAM

Tsinghua University and Galaxea AI · 2026

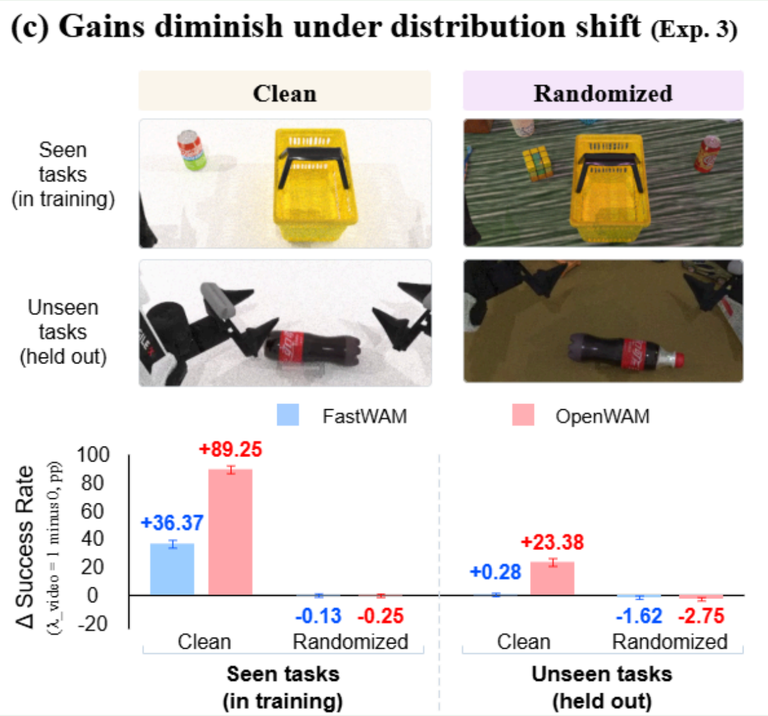
WAM training · Direct VLA action inference

$$\underbrace{\mathcal{L}_{\text{action}} + \mathcal{L}_{\text{video}}}_{\text{training}} \qquad \underbrace{\pi_\theta(A \mid o_t, g_t)}_{\text{deployment}}$$

Input	The current robot observation and instruction.
Output	An action chunk during deployment.
How it works	Train with both video and action targets. Prevent actions from reading future-video tokens, allowing deployment to generate commands directly.

Fast-WAM · Fig. 1 and real-world latency comparison

Does predicting video during training improve action?



Controlled training ablation · Fig. 1c · clean and visually changed scenes

Train two versions

Use the same architecture, starting weights, robot data and number of updates. Switch the future-video loss on or off.

Run both policies

In RoboTwin 2.0, both tested policy architectures improve on familiar, clean tasks. The advantage is inconsistent under visual changes.

LingBot-VA

Robbyant, Ant Group · 2026

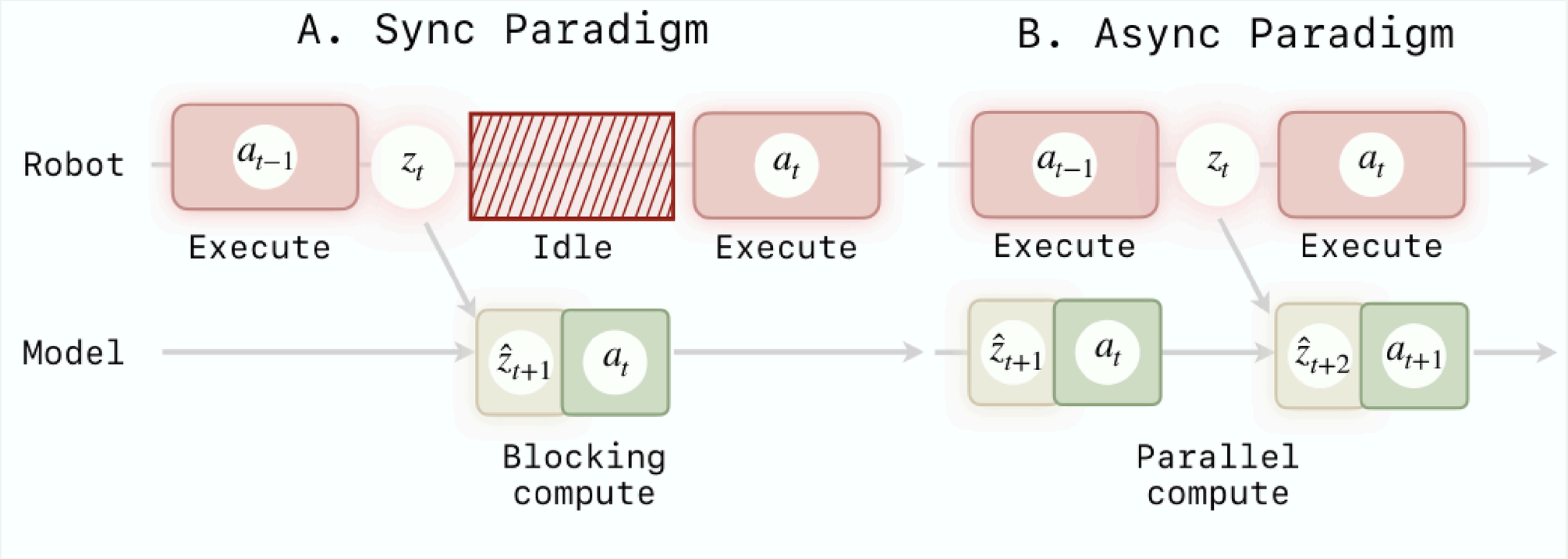
WAM · Causal video-action generation for closed-loop control

$$p_\theta(a_{t:t+H-1}, o_{t+1:t+H} \mid o_{\le t}, a_{<t}, g_t)$$

Input	Visual and action history, robot state and the task instruction.
Output	The next action chunk together with its visual forecast.
How it works	Couple video and action experts using causal attention. Feed back real observations and overlap prediction with motor execution.

LingBot-VA · Sec. 3.4 and Fig. 4 · Robbyant · LingBot-VA · author video

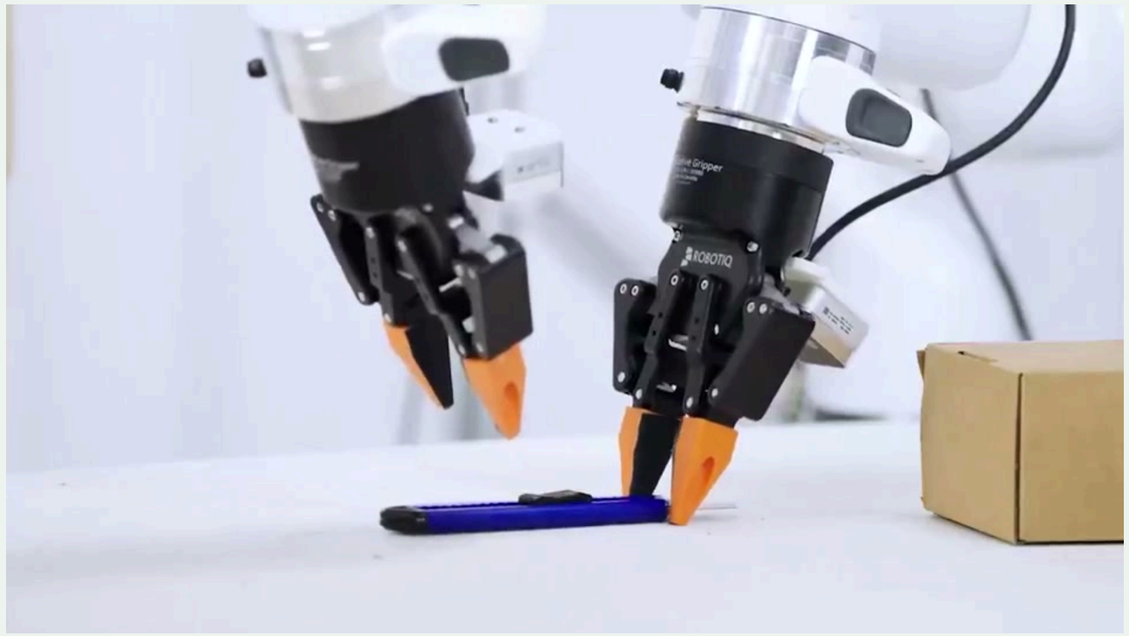
LingBot-VA computes the next chunk while the robot moves



LingBot-VA · Fig. 4, panels A and B · the two rows show the robot and model on the same clock.

Overlap prediction and execution. Before issuing the next chunk, use a fresh observation and the executing action to update the forecast.

What if contact changes while the model is computing?



Robbyant · LingBot-VA · physical manipulation from the authors' demo

Watch the fingers

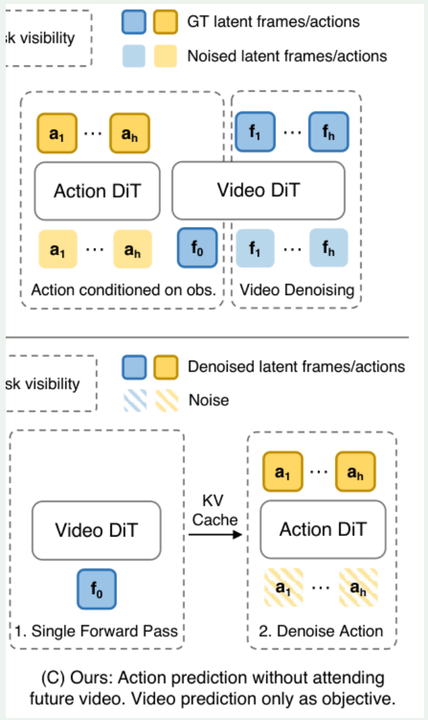
Pause just before contact. A prediction made earlier could put the object at a slightly different position.

Which frame should guide the next move?

Use the latest camera evidence available to the controller. Check whether the next command corrects the error.

Robbyant · LingBot-VA · author video · LingBot-VA · Sec. 3.4 and Fig. 4

Fast-WAM keeps video prediction in training



Fast-WAM · Fig. 1c · training above, inference below

Train on both targets

The same current observation supports two predictions, future video and robot commands. The action stream cannot read the future-video tokens.

Generate only the commands

At deployment the model encodes the current view once and skips future-video denoising. One action chunk took 190 ms on the authors' towel-folding setup, versus 810 ms for their imagine-then-act variant.

Fast-WAM · Fig. 1 and real-world latency comparison

The case for video world models in robotics

Learn from video	Reuse visual experience beyond robot demonstrations.
Make a visual plan	Predict intermediate scenes that guide the robot's actions.
Simulate interaction	Evaluate policies and create training demonstrations.

DreamZero · paper · VERA · MIT CSAIL · DreamDojo · project

The cost of generating video

Latency	The world keeps moving while the model predicts.
Prediction error	A realistic image can show the wrong physical outcome.
Unnecessary detail	Points or features may be enough to choose an action.

Fast-WAM · Fig. 1 and real-world latency comparison · LingBot-VA · Sec. 3.4 and Fig. 4 · Huang et al. · PointWorld · Fig. 2 and planning protocol

Training and deployment are different choices

TRAINING

Learn from future video.

≠

DEPLOYMENT

Generate video before acting.

Fast-WAM trains with video prediction and acts without generating future video.

Fast-WAM · Fig. 1 and real-world latency comparison · What Does World Modeling Actually Improve in Robot Action Learning? · Fig. 1

When is video generation worth it?

Higher task success.

Better recovery.

At the same decision-time budget.