WM@BOOTH 2026 · CHICAGO

AdaJEPA: An Adaptive Latent World Model

Ying Wang[1,2] · Oumayma Bounou[1] · Yann LeCun[1,2] · Mengye Ren[1] [1]New York University [2]AMI Labs

agenticlearning.ai/adajepa

Most WMs are frozen at test time

When predictions go wrong, MPC optimizes the wrong objective.

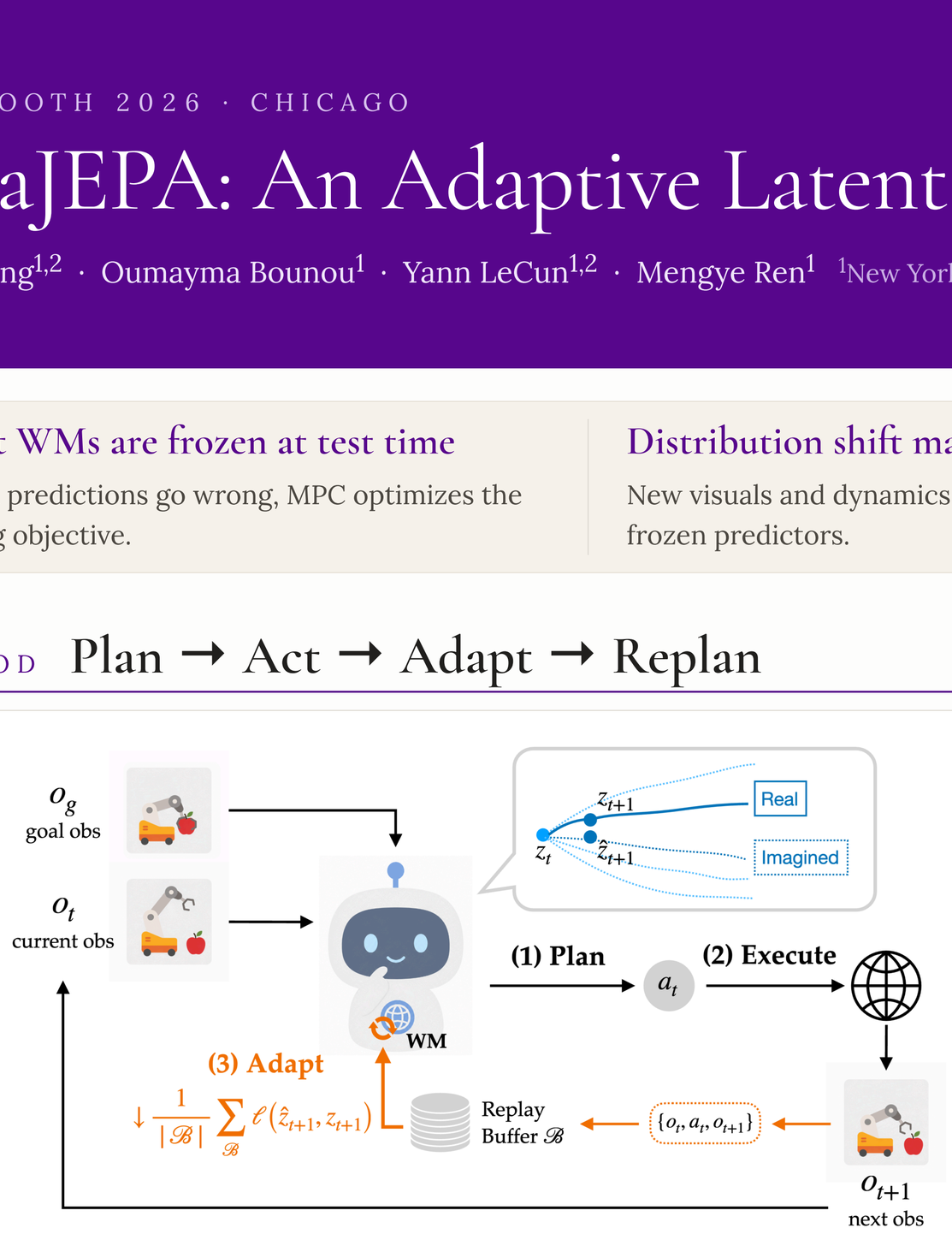
Distribution shift makes it worse

New visuals and dynamics will break frozen predictors.

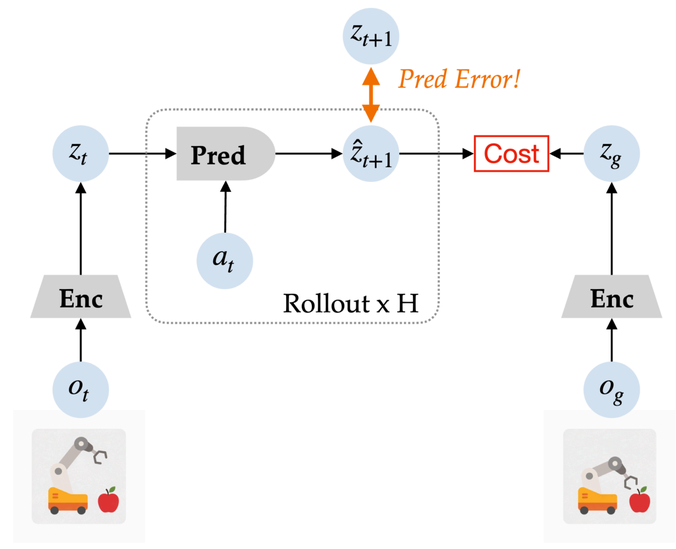
Every execution is free supervision

Each observed transition is a self-supervised target — adapt on it.

METHOD Plan → Act → Adapt → Replan



Test-time adaptation inside the closed MPC loop: execute, observe, adapt, replan.



Latent goal-reaching MPC with a JEPA world model.

ADAPTATION LOSS

$$\mathcal{L}_{ada} = \ell(\, f_\theta(z_t, a_t),\ sg(z_{t+1})\,)$$

DEFAULTS

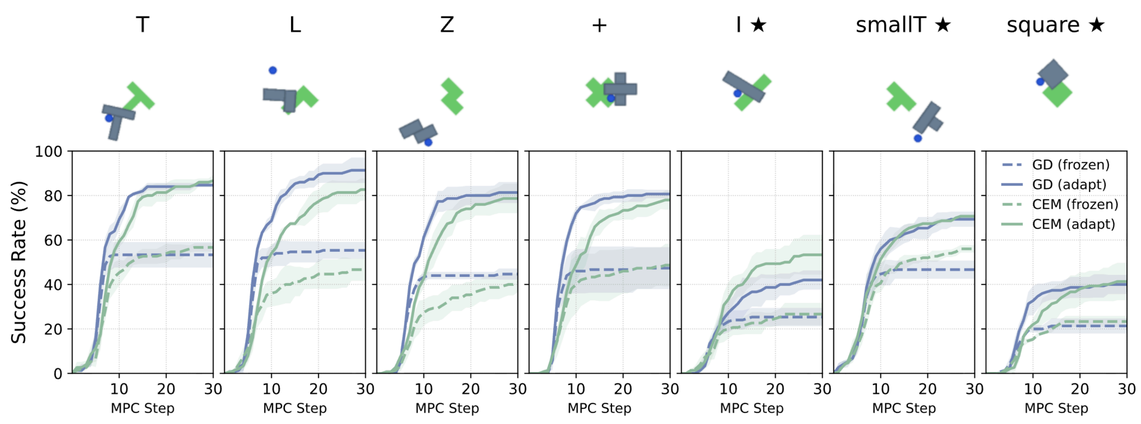
1 gradient step per replan · same lr as training · last layers of enc and pred only · buffer of 5 transitions

COST

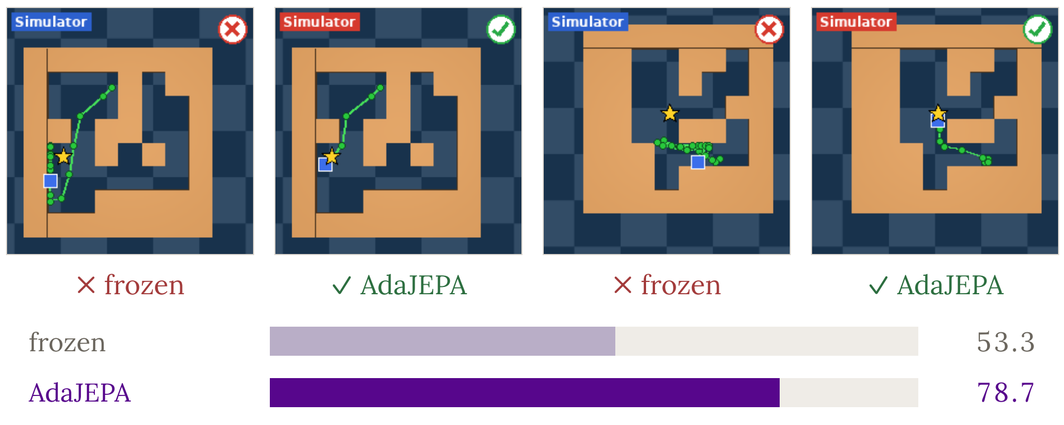
no rewards, no demos · ≤ +0.03 s per replan

RESULTS Consistent gains under every shift, from a single adaptation step per replan

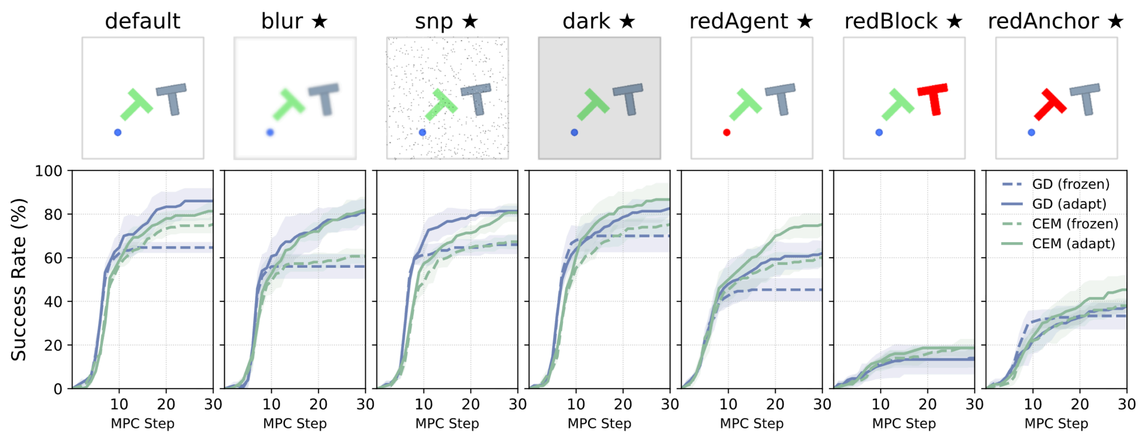
Shape shifts — PushObj, ★ = unseen shapes.



Layout shifts — held-out maze layouts



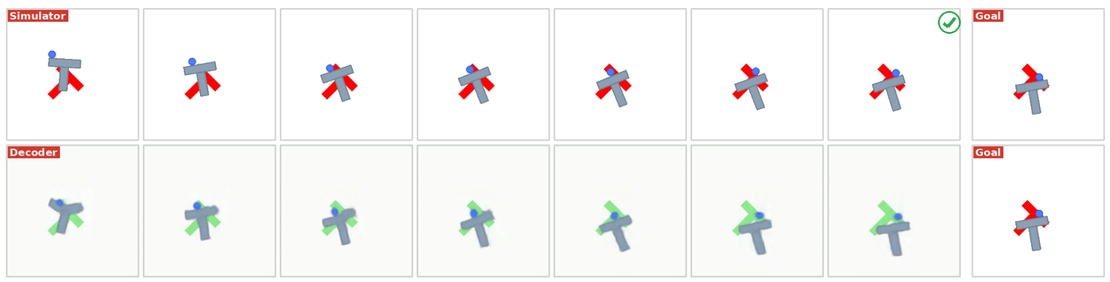
Visual shifts — blur, noise, lighting, and color changes on PushT.



Dynamics shifts — unseen object density and damping at test time.

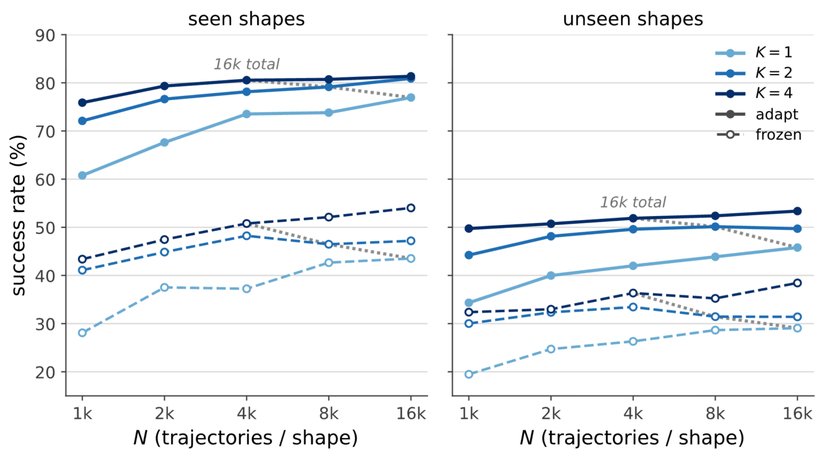
bars: frozen (gray) vs AdaJEPA (purple), success % with GD planning

What does adaptation change? — decoded rollouts stay on the learned manifold.



An unseen red block is decoded as the training-domain gray one — adaptation recalibrates predictions rather than relearning features.

Data efficiency — adaptation compensates for small training sets.



28 → 61%

in the low-data regime — beating a frozen model trained on 16× more data

Takeaway World models should not stay frozen after training. One self-supervised gradient step per replan — on the transition the agent just caused — recovers planning under shift, at negligible cost, for any JEPA world model.