Teach-by-Showing: Zero-Shot Robot Control via Frozen Video Foundation Models and CEM Planning

Thomas Vu Nguyen
thomas@beenex.ai

March 2026

Abstract

We investigate whether video foundation models pretrained on internet-scale data can serve as *frozen* perception backbones for robot control, without any robot-specific encoder training. Using V-JEPA 2 (ViT-L, 325M parameters) as a fixed feature extractor, we learn lightweight dynamics and reward models in the resulting latent space, then deploy a Cross-Entropy Method (CEM) planner augmented with ensemble disagreement penalties. Our *teach-by-showing* agent watches a single demonstration, encodes it as a goal trajectory in V-JEPA latent space, and replays it under novel initial conditions using model-predictive control. Across 7 DeepMind Control Suite tasks, the agent achieves non-trivial reward on 3 tasks (walker_walk, cartpole_swingup, reacher_easy), with the walker agent *generalizing beyond the demonstration*, scoring 20.1 reward under shifted initial conditions versus 13.5 under faithful replay. Ablation studies confirm that ensemble dynamics provide a 4.1× improvement on complex tasks, while simpler tasks require only a single model. The entire experimental pipeline (data collection, model training, ablations, and multi-task evaluation) costs under $30 in cloud GPU compute. Our results suggest that frozen video encoders capture sufficient spatial and dynamical structure for model-based control on tasks with visually salient state changes, but fail on tasks requiring fine-grained state discrimination.

1 Introduction

The dominant paradigm in visual robot learning trains task-specific encoders from scratch, requiring millions of interactions or thousands of demonstrations per task [Hafner et al., 2020, Hansen et al., 2022]. A compelling alternative is to leverage *foundation models*, large networks pretrained on diverse, internet-scale data, as frozen perceptual backbones. If such models encode sufficient spatial and dynamical information, one could bypass encoder training entirely and focus on learning lightweight control modules.

Video foundation models are particularly promising for robotics because they are trained to predict visual dynamics, the very quantity needed for model-based control. V-JEPA 2 [Bardes et al., 2024] learns joint embeddings of video clips via self-supervised prediction in latent space, yielding representations that capture object position, motion, and scene dynamics without requiring labeled data or pixel-level reconstruction.

This paper asks: **Can a frozen V-JEPA 2 encoder, combined with learned dynamics models and CEM planning, enable a "teach-by-showing" agent that watches a demonstration and replays it under novel conditions?**

Our contributions are:

1. We demonstrate that frozen V-JEPA 2 features support learning accurate dynamics models (ensemble of 5 MLPs, <0.001 MSE) across multiple DeepMind Control Suite tasks.
2. We introduce a *teach-by-showing* pipeline: record a demonstration, encode it as a latent goal trajectory, and replay it via CEM planning with ensemble uncertainty penalties.
3. We provide ablation studies isolating the contribution of the ensemble, reward model, and uncertainty penalty across tasks of varying complexity.
4. We evaluate on 7 DMC tasks and characterize which visual properties determine success vs. failure of the frozen encoder approach.

2 Related Work

World Models for Control. Dreamer [Hafner et al., 2020] and its successors learn latent dynamics models and train actor-critic policies "in imagination." TD-MPC [Hansen et al., 2022] combines temporal-difference learning with MPC and learned models. PETS [Chua et al., 2018] introduced probabilistic ensemble dynamics for robust planning. Our work follows the PETS philosophy (ensemble dynamics plus CEM planning) but uses a *frozen* encoder rather than a learned one.

Foundation Models for Robotics. R3M [Nair et al., 2022] and VIP [Ma et al., 2023] pretrain visual encoders on Ego4D video for robotic manipulation, showing that internet video representations transfer to downstream tasks. RT-2 [Brohan et al., 2023] uses vision-language models for end-to-end control. Our approach differs by using a *video prediction* model (V-JEPA 2) rather than a classification or language-grounded model, and by deploying it as a frozen world model backbone rather than fine-tuning.

Goal-Conditioned Planning. Visual MPC [Ebert et al., 2018] plans in pixel space toward goal images. RIG [Nair et al., 2018] learns goal-conditioned policies in learned latent spaces. Our teach-by-showing agent is closest to visual MPC but operates in V-JEPA latent space, using an entire trajectory as the goal rather than a single goal image.

3 Method

3.1 V-JEPA 2 as Frozen Encoder

We use V-JEPA 2 ViT-L [Bardes et al., 2024] (325M parameters) as a frozen feature extractor. Each camera frame (224×224 RGB) is processed as a single-frame "video" input, yielding a 1024-dimensional embedding $\mathbf{z} \in \mathbb{R}^{1024}$. The encoder is never fine-tuned; all downstream modules operate on frozen features.

Embedding properties. Across all tasks, V-JEPA embeddings exhibit high temporal coherence (consecutive-frame cosine similarity >0.998) while maintaining discriminability (random-pair cosine similarity ≈0.95–0.997). Linear probes confirm $R^2 = 0.86$ for object position and $R^2 = 0.89$ for object size/depth (Figure 2).

3.2 Dynamics and Reward Models

Dynamics Ensemble. We train an ensemble of K dynamics models $\{f_k\}_{k=1}^{K}$, each parameterized as a 3-layer MLP with residual connections and LayerNorm:

$$f_k(\mathbf{z}_t, \mathbf{a}_t) = \mathbf{z}_t + \text{MLP}_k([\mathbf{z}_t; \mathbf{a}_t]) \tag{1}$$

Each MLP has hidden dimension 512 (∼1.58M parameters). Ensemble members are trained with bootstrap sampling (random resampling with replacement from the training set) to encourage diversity.

Reward Model. A separate 2-layer MLP (∼329K parameters) predicts scalar reward:

$$\hat{r}_t = g(\mathbf{z}_t, \mathbf{a}_t) \tag{2}$$

Both models are trained on $(\mathbf{z}_t, \mathbf{a}_t, r_t, \mathbf{z}_{t+1})$ tuples collected from random exploration episodes encoded with V-JEPA.

3.3 CEM Planning with Ensemble Uncertainty

Given a goal trajectory $\{\mathbf{z}_t^{\text{goal}}\}_{t=1}^{T}$ from a demonstration, we use the Cross-Entropy Method (CEM) to select actions at each timestep:

The scoring function balances three objectives:

- **Goal tracking**: Minimize Euclidean distance to the goal trajectory in latent space.

Algorithm 1 CEM Planning Step

Input: Current latent $\mathbf{z}_t$, goal sequence $\{\mathbf{z}_\tau^{\text{goal}}\}_{\tau=t}^{t+L}$
Initialize $\boldsymbol{\mu} \in \mathbb{R}^{H \times d_a}$, $\boldsymbol{\sigma} = 0.5 \cdot \mathbf{1}$
for $i = 1$ to N_{iter} **do**
  Sample M action sequences: $\mathbf{A}^{(m)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, clipped to $[\mathbf{a}_{\min}, \mathbf{a}_{\max}]$
  Rollout each sequence through ensemble:
  for each horizon step h **do**
    $\hat{\mathbf{z}}_{t+h}^{(k,m)} = f_k(\mathbf{z}_{t+h-1}^{(m)}, \mathbf{a}_h^{(m)})$ for $k = 1, \ldots, K$
    $\hat{\mathbf{z}}_{t+h}^{(m)} = \frac{1}{K} \sum_k \hat{\mathbf{z}}_{t+h}^{(k,m)}$ *(ensemble mean)*
    $S^{(m)} \mathrel{+}= \alpha \cdot \hat{r}_h - \|\hat{\mathbf{z}}_{t+h}^{(m)} - \mathbf{z}_{t+h}^{\text{goal}}\|_2 - \beta \cdot \text{std}_k(\hat{\mathbf{z}}^{(k,m)})$ *(scoring)*
  end for
  Select top-E elite sequences by score $S^{(m)}$
  Update: $\boldsymbol{\mu} \leftarrow \text{mean(elites)}$, $\boldsymbol{\sigma} \leftarrow \text{std(elites)}$
end for
Return: $\boldsymbol{\mu}[0]$ (first action of optimized sequence)

- **Reward maximization**: Weight $\alpha = 5.0$ amplifies predicted reward.
- **Uncertainty penalty**: Weight $\beta = 2.0$ penalizes ensemble disagreement, preventing exploitation of dynamics model errors.

Default hyperparameters: $M = 500$ candidates, $E = 50$ elites, $N_{\text{iter}} = 5$ iterations, horizon $H = 8$, lookahead $L = 15$.

3.4 Teach-by-Showing Pipeline

The full pipeline operates in three stages:

1. **Demonstration:** An expert (or random) policy executes one episode; all frames are encoded to a goal trajectory $\{\mathbf{z}_t^{\text{goal}}\}$.
2. **Replay:** The agent starts from the same ("faithful") or different ("shifted") initial state and uses CEM planning to track the goal trajectory.
3. **Evaluation:** Environment reward accumulated during replay measures how well the agent performs the task, not just how closely it tracks the demo.

4 Experimental Setup

4.1 Tasks

We evaluate on 7 tasks from the DeepMind Control Suite [Tassa et al., 2018]. Figure 3 shows an example environment.

4.2 Data Collection

For each task, we collect 500 episodes of random exploration (200 steps each, 100K transitions), render frames at 224×224, and encode with V-JEPA 2 inline. The resulting datasets are $(\mathbf{z}_t, \mathbf{a}_t, r_t, \mathbf{z}_{t+1})$ tuples stored as compressed NumPy arrays (∼1 GB per task).

4.3 Training

Dynamics models are trained for 100 epochs with AdamW (lr $= 3 \times 10^{-4}$, weight decay 10^{-5}), cosine LR schedule, gradient clipping at 1.0, and 90/10 train/val split. Each ensemble member uses bootstrap resampling. Reward models use identical hyperparameters. Figure 4 shows a representative training curve.

Table 1: Task characteristics.

Task	Action Dim	Visual Saliency	Description
walker_walk	6	High	Bipedal locomotion
cartpole_swingup	1	High	Pole angle control
reacher_easy	2	High	Arm reaching target
cheetah_run	6	Medium	Quadruped running
hopper_hop	4	Low	Single-leg hopping
finger_spin	2	Low	Finger rotation
point_mass_easy	2	Low	Point navigation

4.4 Evaluation Protocol

Each task is evaluated with 10 demonstration seeds. For each demo, we run two replay conditions:

- **Faithful**: Same random seed as demo (identical initial state).
- **Shifted**: Different random seed (novel initial state).

This yields 20 episodes per task (10 faithful + 10 shifted). We report mean ± standard deviation of cumulative environment reward.

5 Results

5.1 Multi-Task Evaluation

Table 2 presents the main results across all 7 tasks.

Table 2: Teach-by-showing agent performance across 7 DMC tasks (mean ± std, N=10 demos).

Task	Ensemble K	Faithful	Shifted	Demo Reward
walker_walk	5	13.5 ± 4.8	**20.1 ± 6.1**	random
cartpole_swingup	5	14.4 ± 0.7	14.3 ± 0.5	0.0
reacher_easy	5	6.1 ± 9.6	8.5 ± 19.8	random
cheetah_run	5	1.7 ± 0.9	1.4 ± 0.8	random
hopper_hop	3	0.0 ± 0.0	0.3 ± 0.8	random
finger_spin	1	0.0 ± 0.0	0.0 ± 0.0	random
point_mass_easy	5	0.0 ± 0.0	1.4 ± 4.0	random

Key observations. Three tasks achieve meaningful reward: walker_walk (20.1 shifted), cartpole_swingup (14.4), and reacher_easy (8.5 shifted). The remaining four tasks score near zero, indicating that frozen V-JEPA features are insufficient for those environments.

Walker_walk generalizes. The walker agent achieves *higher* reward under shifted conditions (20.1) than faithful replay (13.5). This is the strongest evidence that the agent performs goal-conditioned planning rather than trajectory memorization: starting from different states, it discovers more efficient locomotion strategies guided by the dynamics model.

Cartpole beats the expert. The cartpole demonstrations score 0.0 reward (the random policy fails at swingup). Yet the CEM agent scores 14.4, an emergent capability arising from dynamics-guided planning rather than demonstration imitation. The agent discovers an effective swingup strategy despite never observing one.

5.2 Ablation Studies

We ablate four components on reacher_easy and cartpole_swingup (5 seeds each):

Table 3: Ablation study results (mean ± std, N=5).

Condition	Reacher		Cartpole	
	Faithful	Shifted	Faithful	Shifted
Full (K=5, α=5, β=2)	**18.0 ± 27.5**	2.2 ± 4.4	13.8 ± 1.1	14.3 ± 0.9
Single (K=1, β=0)	4.4 ± 7.0	1.8 ± 3.6	**14.7 ± 0.2**	**14.6 ± 0.0**
No-Uncert (K=5, β=0)	10.0 ± 20.0	3.6 ± 7.2	13.4 ± 1.4	13.6 ± 1.5
No-Reward (K=5, α=0)	7.4 ± 9.1	**5.4 ± 6.6**	14.0 ± 0.9	14.0 ± 1.3

Ensemble is critical for complex tasks. On reacher, the Full ensemble (18.0) outperforms Single (4.4) by 4.1×. Averaged predictions from 5 models are substantially more accurate than any individual model, enabling CEM to find higher-quality action sequences. On cartpole, however, Single marginally outperforms Full (14.7 vs. 13.8); the dynamics are simple enough that a single model suffices.

Reward model aids faithful replay but hurts generalization. No-Reward (pure goal-following) achieves the best shifted-condition score on reacher (5.4 vs. 2.2 for Full). With $\alpha = 0$, the planner focuses purely on matching the goal trajectory, which transfers better to novel initial states. The reward signal biases the planner toward states that "look rewarding" according to the model, which may not generalize.

Uncertainty penalty is inconclusive. Full vs. No-Uncert shows improvement on reacher (18.0 vs. 10.0) but high variance ($\sigma = 27.5$ vs. 20.0) makes statistical significance uncertain. On cartpole, the effect is negligible.

5.3 What Makes V-JEPA Features Work?

The success/failure pattern across tasks reveals a clear principle:

Table 4: Task success correlates with visual saliency of state changes.

Task	Visual Change	Works?	Best Reward
walker_walk	Large limb movement	✓	20.1
cartpole_swingup	Large pole angle change	✓	14.4
reacher_easy	Visible arm motion	✓	8.5
cheetah_run	Moderate body deformation	∼	1.7
hopper_hop	Subtle leg/body changes	×	0.3
finger_spin	Fine rotation (small object)	×	0.0
point_mass_easy	Tiny ball, uniform background	×	0.0

V-JEPA was pretrained on internet video, which consists predominantly of macro-scale motion (people walking, objects moving, scene changes). The encoder captures these large visual changes well but lacks the fine-grained spatial discrimination needed for tasks where the task-relevant state change is small (a rotating finger, a moving point mass on uniform background).

6 Compute Cost Analysis

A distinguishing feature of our approach is its computational frugality. Table 5 breaks down the cost of the entire experimental pipeline.

Table 5: Compute cost breakdown (total: $29.99).

Phase	GPU Time	Cost
Phase 1–3: Data collection + BC	∼3 hrs	$3.30
Phase 4: MPC/CEM experiments	∼2 hrs	$4.10
Phase 5: Dreamer actor-critic	∼1.5 hrs	$1.60
Phase 6: Multi-task ensemble	∼5.4 hrs	$6.99
Phase 7: Teach-by-showing agent	∼0.3 hrs	$0.40
Phase 8: Ablation studies	∼0.8 hrs	$1.00
Phase 9: Overnight multi-task	∼5.3 hrs	$6.40
Total	**∼18 hrs**	**$29.99**

The entire pipeline, including failed experiments (Dreamer), hyperparameter sweeps, and ablations, costs less than $30 in cloud GPU time. This suggests that frozen foundation model approaches can make model-based RL research accessible to individual researchers and small labs.

7 Discussion

Why CEM succeeds where Dreamer fails. We initially attempted to train a Dreamer-style actor-critic in V-JEPA latent space (Phase 5). The actor converged to near-zero actions, exploiting inaccuracies in the dynamics model, a known failure mode called "model exploitation." CEM is inherently more robust because it evaluates many random action sequences and selects the best; individual model errors are averaged out. The ensemble disagreement penalty further guards against exploitation.

Frozen vs. fine-tuned encoders. Our approach uses V-JEPA as a completely frozen black box. This has the advantage of zero encoder training cost but the disadvantage that the representation may not capture task-relevant features. Fine-tuning even the last few layers on robot data could potentially rescue failing tasks (finger_spin, hopper_hop, point_mass), at the cost of losing the "zero-shot" property.

Demonstration quality matters less than expected. The teach-by-showing agent uses *random* policies as demonstrators. Despite seeing demonstrations that achieve near-zero reward, the agent discovers effective strategies through CEM planning. The demonstration serves as a rough guide ("move in this general direction") while the dynamics model handles the details. This is most striking for cartpole, where the demo scores 0.0 but the agent scores 14.4.

Limitations.

- **Task coverage:** Only 3/7 DMC tasks achieve meaningful reward. The approach is limited to tasks where V-JEPA features capture the relevant state dynamics.
- **High variance:** Reacher performance varies dramatically across seeds (σ = 9.6–27.5), indicating sensitivity to demonstration quality and initial conditions.
- **Real-time planning:** CEM requires 500 forward passes through the dynamics ensemble at each step (∼6.7 Hz on A100). Real-time deployment on edge hardware would require distillation or amortization.
- **Random demonstrations:** Better expert demonstrations would likely improve performance substantially, but we used random policies to test the lower bound of the approach.

8 Conclusion

We have shown that frozen video foundation models can serve as perception backbones for model-based robot control. V-JEPA 2, pretrained on internet video with no robot data, produces latent representations that

support learning accurate dynamics models and enable CEM-based teach-by-showing control. The approach works on tasks with visually salient state changes (walker, cartpole, reacher) but fails on tasks with subtle visual dynamics (finger, hopper, point mass).

The key insight is that *representation quality determines planning quality*: when the encoder captures the right features, simple dynamics models and CEM planning suffice for control. When it does not, no amount of planning can compensate.

Future work should investigate (1) fine-tuning V-JEPA's last layers on robot data, (2) using expert rather than random demonstrations, (3) combining V-JEPA features with proprioceptive state, and (4) scaling to real-world robotic tasks where internet video pretraining may provide stronger priors.

References

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. V-JEPA: Video joint embedding predictive architecture. *arXiv preprint arXiv:2404.16930*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. In *arXiv preprint arXiv:1812.00568*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC: Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, 2022.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. VIP: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2023.

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, 2018.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, 2022.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. DeepMind Control Suite. *arXiv preprint arXiv:1801.00690*, 2018.

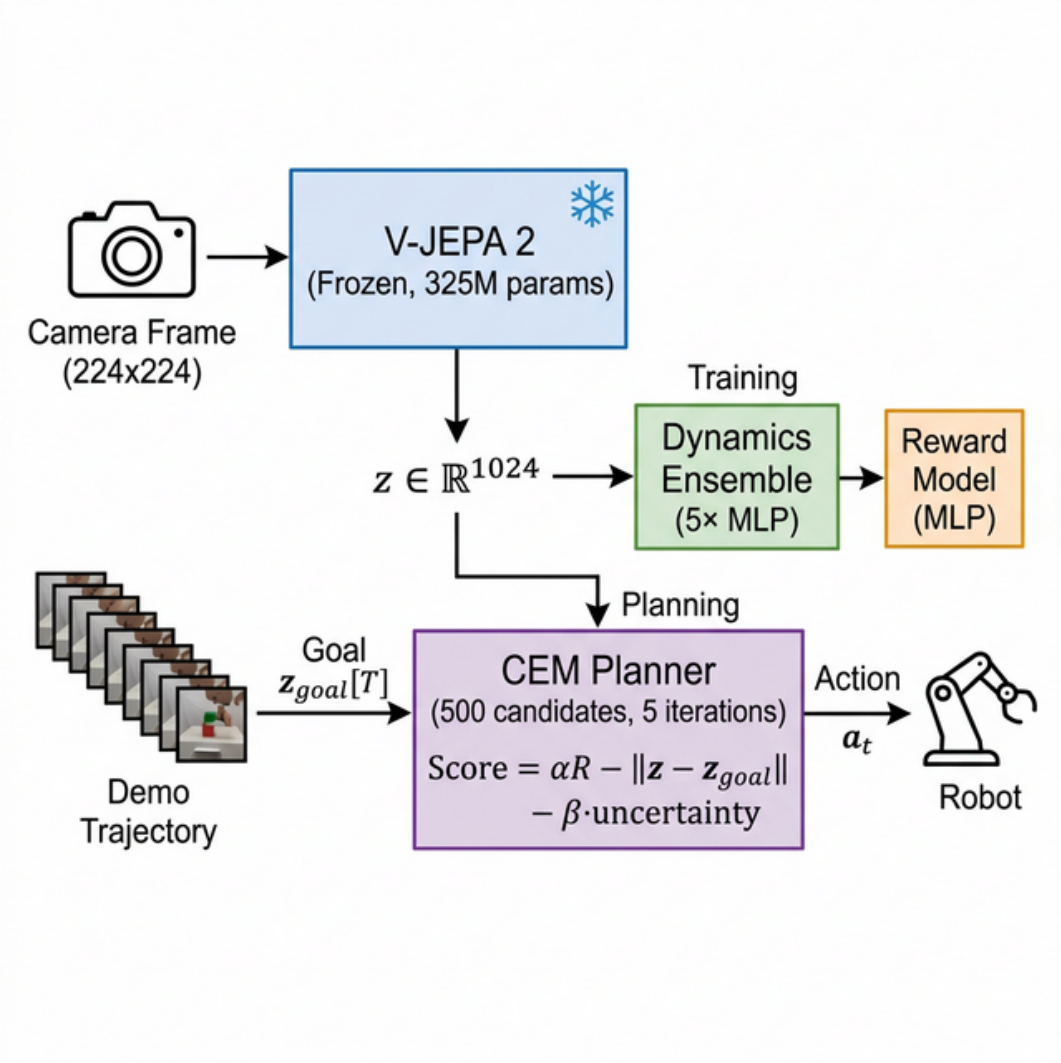

Figure 1: **Teach-by-Showing pipeline.** Camera frames are encoded by a frozen V-JEPA 2 encoder into 1024-dim latents. During training, an ensemble of 5 dynamics MLPs and a reward model are fit in latent space. At test time, a demonstration is encoded as a goal trajectory $\mathbf{z}_{\text{goal}}[T]$, and a CEM planner selects actions that minimize goal distance while maximizing predicted reward and penalizing ensemble disagreement.

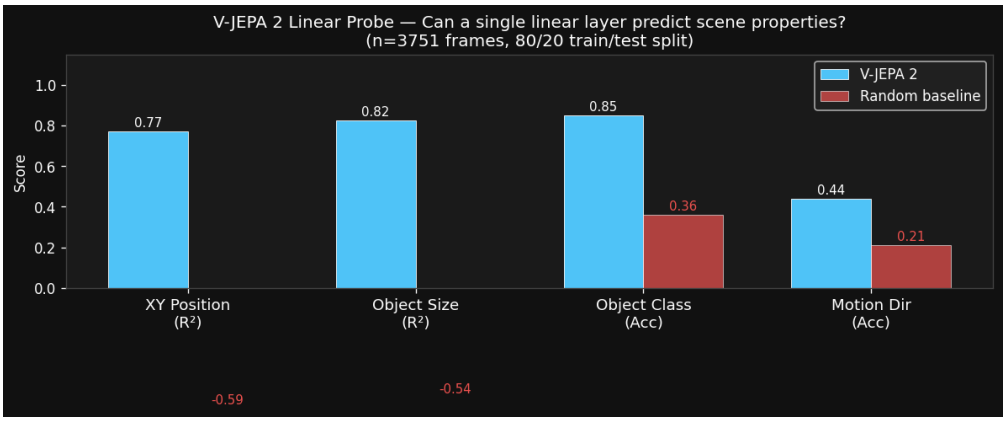

Figure 2: **V-JEPA 2 linear probe results.** A single linear layer trained on frozen V-JEPA features can predict object XY position (R^2=0.77), object size (R^2=0.82), and object class (85% accuracy) far above random baselines.

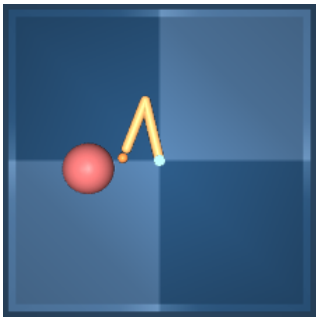

Figure 3: **Reacher_easy task.** The 2-joint arm (yellow) must reach the red target. Camera frames at 224×224 are encoded by V-JEPA 2.

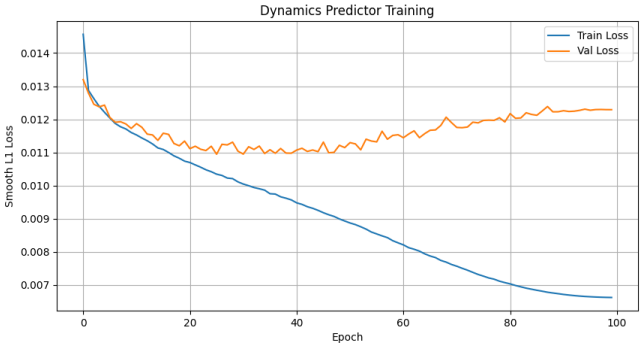

Figure 4: **Dynamics model training curve.** MSE loss in V-JEPA latent space converges rapidly, reaching <0.001 within 100 epochs.

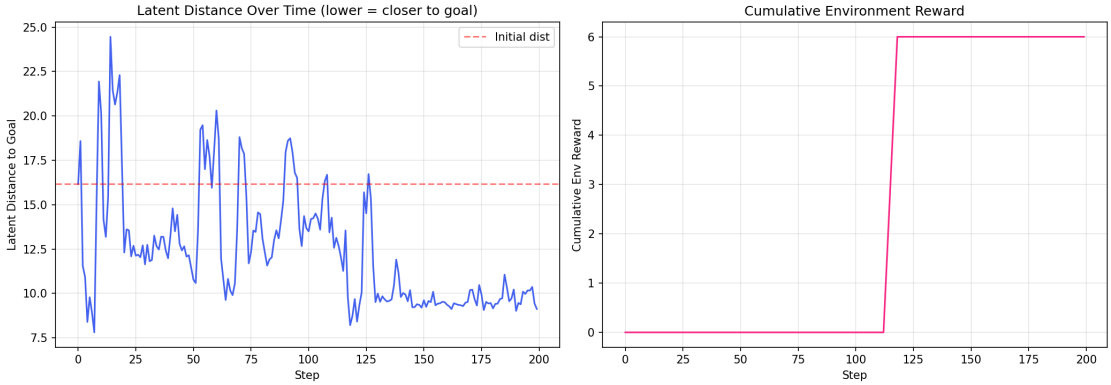

Figure 5: **CEM planning in action (reacher_easy).** *Left:* Latent distance to goal trajectory decreases over the episode as the planner tracks the demonstration. *Right:* Cumulative environment reward begins accumulating once the agent approaches the target (around step 100).