

On-Policy Self-Distillation for Cross-Embodiment Transfer in World Action Models

2026-03-04

1. Problem Statement

World Action Models (WAMs) such as DreamZero jointly predict future video and robot actions through a shared backbone, achieving strong generalization on manipulation tasks. A key promise of WAMs is *cross-embodiment transfer*: leveraging abundant video demonstrations from other robots or humans to improve a target robot’s policy. However, current approaches use cross-embodiment data in a limited way—training only a video prediction objective on non-target data, discarding the action head entirely. The model learns *what the world looks like* when a task is performed but never explicitly learns *how the target robot should act* given what another embodiment demonstrated.

This is a surprisingly impoverished learning signal. The shared video-action backbone already encodes correspondences between visual dynamics and motor commands for the target embodiment. Cross-embodiment demonstrations provide strong constraints on the desired visual trajectory, grasp strategy, and timing. But no training signal connects these two sources of knowledge for action prediction.

We propose **OPSD-CE** (On-Policy Self-Distillation for Cross-Embodiment transfer), a method that uses cross-embodiment demonstrations as *privileged information* to generate dense, per-timestep action supervision for the target embodiment—then distills this knowledge into a student that requires no demonstrations at inference time.

2. Background

2.1 DreamZero and World Action Models

DreamZero is a 14B-parameter WAM built on a Diffusion Transformer (DiT) backbone that jointly predicts video tokens and action tokens via flow matching. Given a task instruction x and current observation s_0 , the model autoregressively generates future video frames $\hat{v}_{1:T}$ and extracts actions $a_{1:T}$ through an implicit inverse dynamics process. The video prediction acts as a visual planner: the model first “imagines” the future, then derives motor commands consistent with that imagined future.

For cross-embodiment data (e.g., human demonstrations, other robot platforms), DreamZero trains only the video prediction loss:

$$\mathcal{L}_{\text{cross}} = \mathcal{L}_{\text{video}}(\theta; x, v^{\text{demo}})$$

discarding the action loss since action labels are unavailable or incompatible. This yields a 42% improvement with 10–20 minutes of cross-embodiment video, but leaves substantial information unused.

Key observation: DreamZero’s authors identify that the primary failure mode is *video generation errors* propagating to incorrect actions—not action extraction failures. The action head is capable when given good visual plans, but the visual planner sometimes errs.

2.2 On-Policy Self-Distillation (OPSD)

OPSD introduces a framework for distilling privileged information into a student policy through on-policy rollouts. A teacher $\pi_{\theta}^T(y \mid x, I)$ conditions on privileged information I unavailable at test time, while a student $\pi_{\phi}^S(y \mid x)$ sees only the input x . The training procedure:

1. The student generates rollouts from its own policy.
2. At each token position along the rollout, the teacher provides a target distribution conditioned on I .
3. The student minimizes per-token divergence against the teacher’s distribution.

OPSD achieves 4–8× token efficiency over GRPO by replacing sparse end-of-sequence rewards with dense per-token supervision. The on-policy rollouts avoid distribution mismatch between training and inference.

3. Method: OPSD-CE

3.1 Setup

Let π_θ denote the shared WAM with parameters θ . We define two operating modes of the *same model*, controlled by a gating scalar $\alpha \in [0, 1]$:

$$\textbf{Teacher: } \pi_\theta^T(a_t \mid s_t, x, v^{\text{demo}}; \alpha = 1)$$

$$\textbf{Student: } \pi_\theta^S(a_t \mid s_t, x; \alpha = 0)$$

where v^{demo} is a cross-embodiment demonstration video (privileged information), s_t is the state at timestep t , and x is the task instruction.

No additional parameters. The cross-embodiment demonstration v^{demo} is tokenized by the existing video tokenizer and injected into the DiT backbone via cross-attention, gated by α . When $\alpha = 0$, the demonstration tokens are zeroed out and the model operates as the student. When $\alpha = 1$, the model has full access (teacher mode). Both modes share all parameters θ .

3.2 Why the Teacher Can Rationalize

The teacher is not an oracle—it must *infer* appropriate target-embodiment actions from a cross-embodiment demonstration. We argue this is tractable because:

1. **Shared visual dynamics.** The demonstration constrains the desired visual trajectory (object motion, contact timing, spatial path). These are largely embodiment-invariant.
2. **Known target-embodiment mapping.** From target-embodiment data $\mathcal{D}_{\text{tgt}}$, the model has learned the mapping from visual trajectories to motor commands for the target robot.
3. **Narrowed search space.** Instead of searching over all motor programs that achieve the goal, the teacher only needs to find commands that reproduce the demonstrated visual trajectory on the target embodiment.

The demonstration acts as a strong prior that collapses the ambiguity inherent in instruction-only conditioning.

3.3 Training Algorithm

Algorithm: OPSD-CE

Input: Shared WAM π_θ , target data $\mathcal{D}_{\text{tgt}}$, cross-embodiment data $\mathcal{D}_{\text{cross}}$, KL coefficient β , learning rate η

Initialize: θ from pre-trained WAM

Repeat until convergence:

Standard WAM training on target-embodiment data:

1. Sample $(x, v_{\text{tgt}}, a_{\text{tgt}}) \sim \mathcal{D}_{\text{tgt}}$
2. $\mathcal{L}_{\text{video}} \leftarrow \text{FlowMatch}(\pi_\theta, x, v_{\text{tgt}})$
3. $\mathcal{L}_{\text{action}} \leftarrow \text{FlowMatch}(\pi_\theta, x, a_{\text{tgt}})$

Cross-embodiment self-distillation:

4. Sample $(x, v^{\text{demo}}) \sim \mathcal{D}_{\text{cross}}$

Phase 1 — Student rollout (on-policy):

5. Roll out $\tau^S = (s_0, a_0^S, s_1, a_1^S, \dots, s_T)$ using $\pi_\theta^S(\cdot \mid s_t, x; \alpha = 0)$

Phase 2 — Dense teacher supervision:

6. **For** each timestep t in τ^S :
 - Sample flow timestep $t_{\text{flow}} \sim \mathcal{U}[0, 1]$, noise $\epsilon \sim \mathcal{N}(0, I)$
 - Teacher velocity: $\mathbf{v}_t^T \leftarrow f_\theta(s_t, x, v^{\text{demo}}, \epsilon, t_{\text{flow}}; \alpha = 1)$
 - Student velocity: $\mathbf{v}_t^S \leftarrow f_\theta(s_t, x, \epsilon, t_{\text{flow}}; \alpha = 0)$

Phase 3 — Distillation loss:

7. $\mathcal{L}_{\text{distill}} \leftarrow \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbf{v}_t^S - \text{sg}(\mathbf{v}_t^T)\|^2$

Joint update:

8. $\mathcal{L} \leftarrow \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}} + \beta \cdot \mathcal{L}_{\text{distill}}$
9. $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

Return: π_θ^S (student mode, $\alpha = 0$)

3.4 Design Decisions

Flow-matching MSE instead of KL. Since DreamZero uses flow matching for action prediction, the teacher and student outputs are velocity fields over continuous action space. Matching velocity fields via MSE:

$$\|\mathbf{v}_t^S - \mathbf{v}_t^T\|^2$$

is equivalent to minimizing the KL divergence between the implied flow distributions, and integrates natively with DreamZero’s existing training infrastructure.

Stop-gradient on teacher (sg). We apply stop-gradient to the teacher’s velocity field to stabilize training, following OPSD. The teacher still improves through $\mathcal{L}_{\text{video}}$ on cross-embodiment data and $\mathcal{L}_{\text{action}}$ on target data. An ablation without stop-gradient (joint gradient flow through both modes) may benefit early training when the teacher’s rationalizations are poor.

On-policy rollouts. The student generates its own trajectories, and the teacher provides corrections at the states the student actually visits. This avoids the distribution mismatch problem of off-policy distillation: if the student drifts into an unusual state, the teacher corrects it *there*, not at states from an expert trajectory the student may never reach.

Single model, gated privileged information. Using a single model with α -gating rather than separate teacher/student networks means: (a) no extra memory or parameters, (b) improvements to the backbone benefit both modes simultaneously, and (c) the framework is compatible with π -Distill-style joint training schedules that anneal α over time.

4. Experimental Plan

4.1 Setup

- **Target embodiment:** AgiBot (7-DoF manipulator)
- **Cross-embodiment sources:** Human demonstrations (video only), YAM robot data (video only), other robot platforms
- $\mathcal{D}_{\text{tgt}}$: AgiBot teleoperation data (scarce: 10–60 min per task)
- $\mathcal{D}_{\text{cross}}$: Human/YAM video demonstrations (abundant: hours of video)

4.2 Baselines

1. **DreamZero (video-only):** Standard co-training with $\mathcal{L}_{\text{video}}$ on cross-embodiment data. No action supervision from $\mathcal{D}_{\text{cross}}$.
2. **Off-policy distillation:** Teacher generates target-embodiment actions from demos; student imitates via SFT on teacher-generated data (not on-policy).
3. **Target-only:** Train only on $\mathcal{D}_{\text{tgt}}$, no cross-embodiment data. Lower bound.
4. **Action co-training:** Train action loss on all embodiments with a learned action-space mapping. Upper bound (requires action labels for all embodiments).

4.3 Key Experiments

Experiment 1: Cross-embodiment transfer efficacy. Compare OPSD-CE vs. baselines on held-out tasks that *only appear in cross-embodiment data* (never demonstrated on AgiBot). This is the cleanest test: can the model do on AgiBot what it only saw a human do?

Experiment 2: Data efficiency scaling. Vary the ratio $|\mathcal{D}_{\text{tgt}}|/|\mathcal{D}_{\text{cross}}|$. Hypothesis: OPSD-CE shows the largest gains when target-embodiment data is most scarce, precisely the regime that matters in practice.

Experiment 3: Embodiment gap. Test across embodiment pairs with varying morphological similarity: AgiBot – YAM (similar grippers), AgiBot – Human (different end-effectors), AgiBot – bimanual robot (different kinematics). Hypothesis: performance degrades gracefully as the embodiment gap widens.

Experiment 4: Ablations.

- On-policy vs. off-policy distillation
- Stop-gradient vs. joint gradient flow
- Full demo video vs. keyframes vs. trajectory sketch as privileged information
- Sensitivity to β (distillation coefficient)

4.4 Evaluation Metrics

- Task success rate on held-out tasks (primary)
- Success rate on cross-embodiment-only tasks (zero-shot transfer)
- Sample efficiency: success rate as a function of target-embodiment data quantity
- Video-action alignment: how well predicted actions match generated video plans

5. Expected Contributions

1. **A principled framework** for using cross-embodiment demonstrations as privileged information in WAMs, moving beyond video-only training objectives.
2. **Dense action supervision from unlabeled video:** cross-embodiment demonstrations that lack target-embodiment action labels are transformed into per-timestep action training signals via self-distillation.
3. **On-policy training for robustness:** corrections occur at states the student actually visits, avoiding the compounding errors of off-policy approaches.
4. **Zero additional parameters:** the method uses a single model with gated privileged information, requiring no architectural changes to the WAM.

The broader significance is that OPSD-CE turns any cross-embodiment video demonstration into a source of dense action supervision for a target robot—a capability that scales naturally with the growing availability of internet-scale manipulation video.

6. Connections to Prior Work

Privileged information distillation. Learning Using Privileged Information (LUPI) trains a teacher with access to information unavailable at test time, then distills to a student. In robotics, this has been applied with access to simulator state, but not with cross-embodiment demonstrations as the privileged modality.

Cross-embodiment transfer. Existing approaches include domain randomization, embodiment-invariant representations, and video-only co-training (as in DreamZero). OPSD-CE is distinct in that it generates *action-level* supervision from cross-embodiment video, rather than relying solely on shared visual representations.

Self-distillation for reasoning. OPSD and π -Distill show that dense self-distillation from privileged information dramatically improves sample efficiency in language model reasoning. We adapt this insight to the visuomotor domain, where the “reasoning” is motor planning and the “privileged information” is a demonstration from another embodiment.

7. Timeline

Phase	Duration
Implementation of α -gated privileged injection in DiT backbone	2 weeks
On-policy rollout infrastructure + distillation loss	2 weeks
Experiment 1–2: Transfer efficacy and data efficiency	3 weeks
Experiment 3–4: Embodiment gap and ablations	3 weeks
Writing and submission	2 weeks