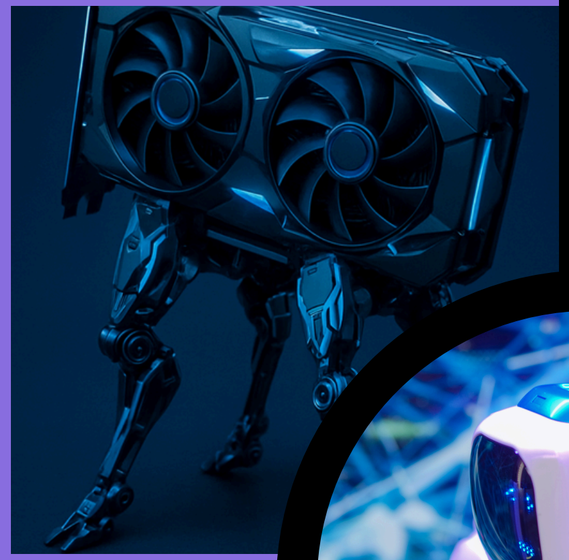
DAYDREAMER: WORLD MODELS FOR PHYSICAL ROBOT LEARNING

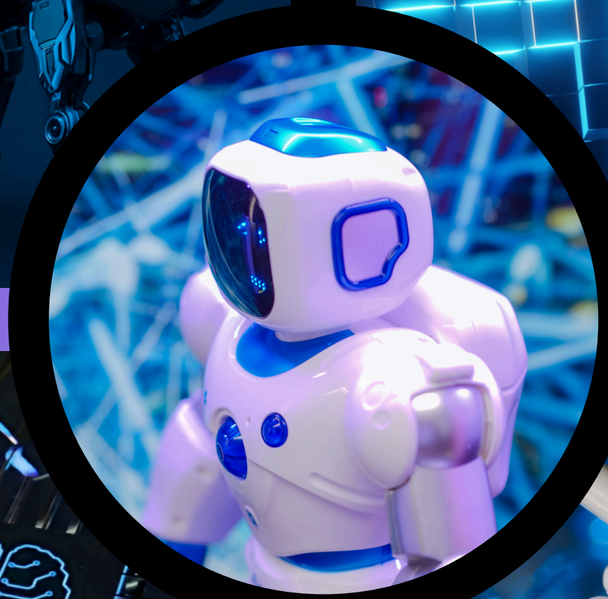
EVALUATION OF VISUAL GENERALIZATION

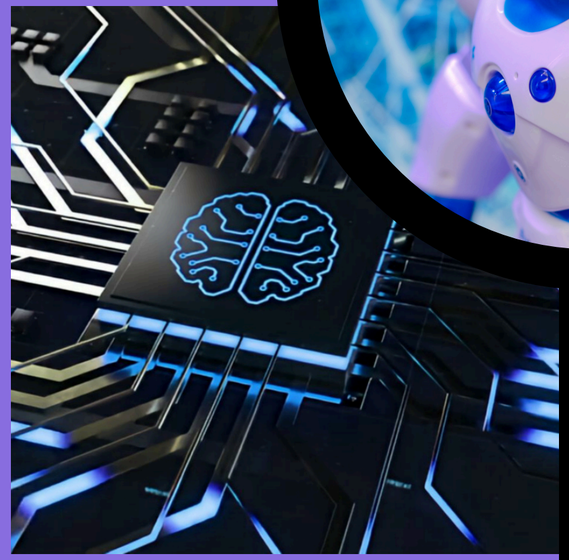
BASED ON WU ET AL. (2022)

GÖRKEM BASAR ,
JOSEPHINE STÖHR,
SINA KHUDIDA

SEMINAR: MACHINE LEARNING IN ROBOTICS
BY PROF. DR. VERENA HAFNER







Motivation

Real-world robot learning is expensive and slow

- **Physical interaction costs time, wear and human supervision**

Pure reinforcement learning needs too much trial-and-error

- **Millions of steps are often required → unrealistic on real robots**

Simulators are an imperfect solution

- **Hard to model real-world physics, perception and variability**
- **Learned behaviors often fail to transfer (sim-to-real gap)**

World models offer a promising alternative

- **Learn environment dynamics from real data**
- **Enable “learning by imagination” instead of physical trial-and-error**

Key open question

- **Do learned world models generalize beyond the exact conditions they were trained on?**
- **Especially: visual and perceptual changes**

DayDreamer
(Wu et al., 2022)

A model-based reinforcement learning algorithm

- **Designed for learning directly on physical robots**
- **No simulator required**

Core idea: learn a world model

- **Predicts how the environment evolves given actions**
- **Trained from real robot experience**

Two main components

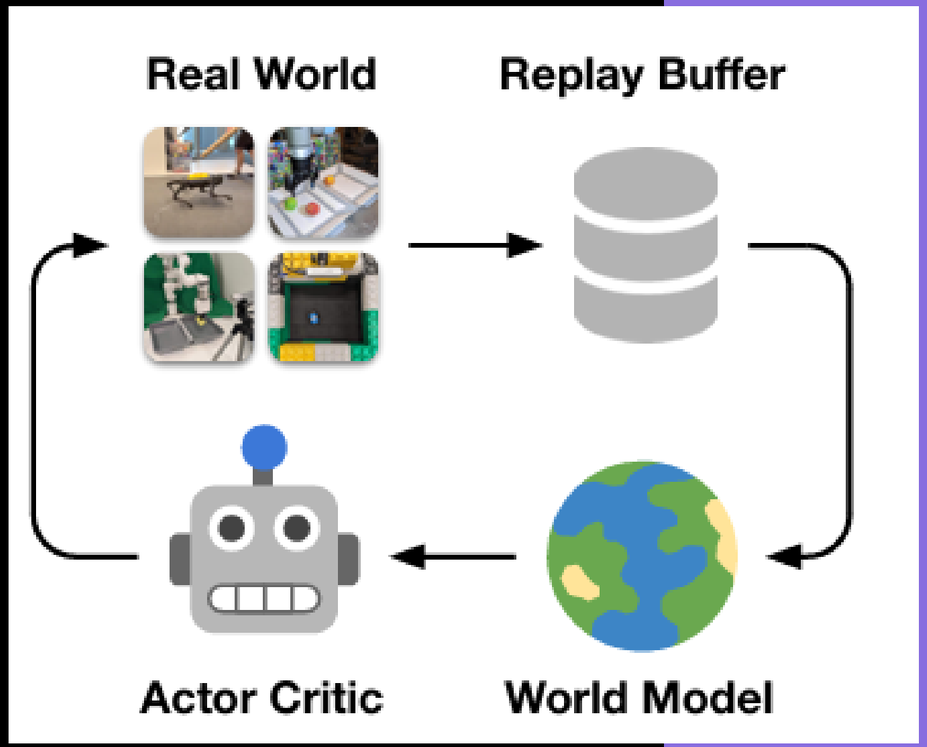
- **World Model: learns dynamics, rewards and latent representations**
- **Actor–Critic: learns behavior using imagined rollouts inside the world model**

Learning by imagination

- **Policies are optimized in the latent space of the world model**
- **Greatly reduces the need for real-world interaction**

Key contribution of the paper

- **Demonstrates that Dreamer can learn locomotion, manipulation, and navigation**
- **Directly in the real world, with high sample efficiency**



Source: Wu et al., CoRL 2022

DayDreamer World Model Architecture

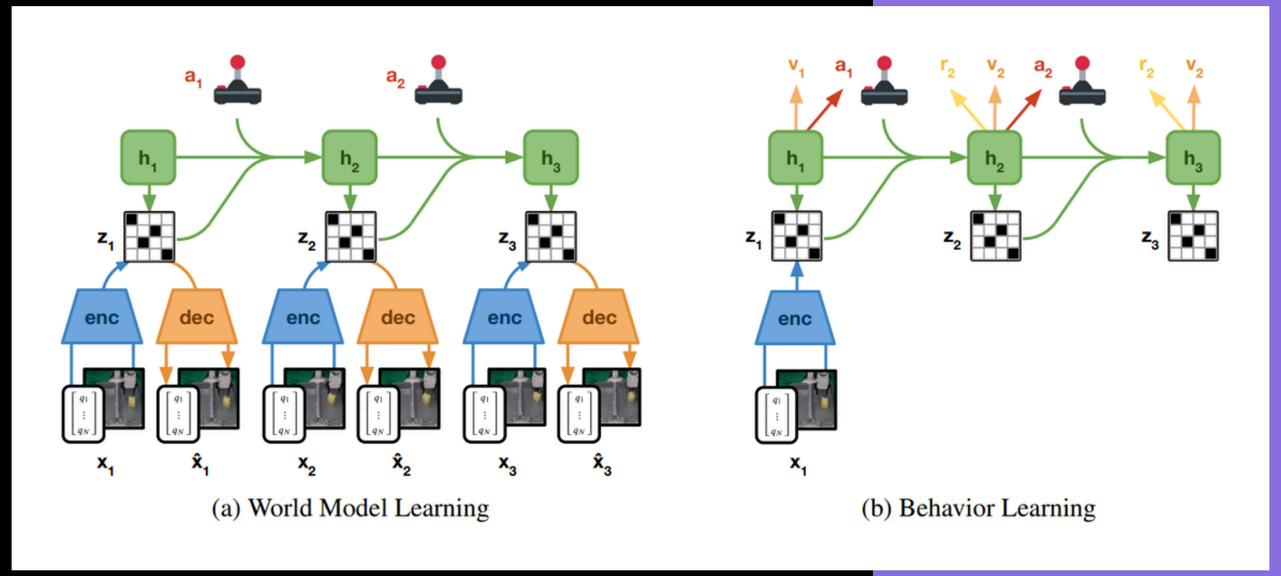
1. Encoder (Perception → Latent)
   - Encodes observations (e.g. images, proprioception) into a compact latent state
   - Learns what aspects of perception are predictive for the future

2. Recurrent State-Space Model (RSSM)
   - Combines:
     - deterministic hidden state (memory)

stochastic latent state (uncertainty / abstraction)
   - Predicts latent transitions conditioned on actions

3. Prediction Heads (Latent → Outputs)
   - Reconstruct observations
   - Predict reward
   - Predict episode termination

4. Imagination-Based Actor–Critic
   - Actor proposes actions in latent space
   - World model rolls out imagined trajectories
   - Critic evaluates returns without real environment interaction



Source: Wu et al., CoRL 2022

DayDreamer: Perception and Domain Transfer

Custom DMC Walker experiments (side vs top viewpoint)
Training curves, evaluation, and transfer analysis

Goal and Hypothesis

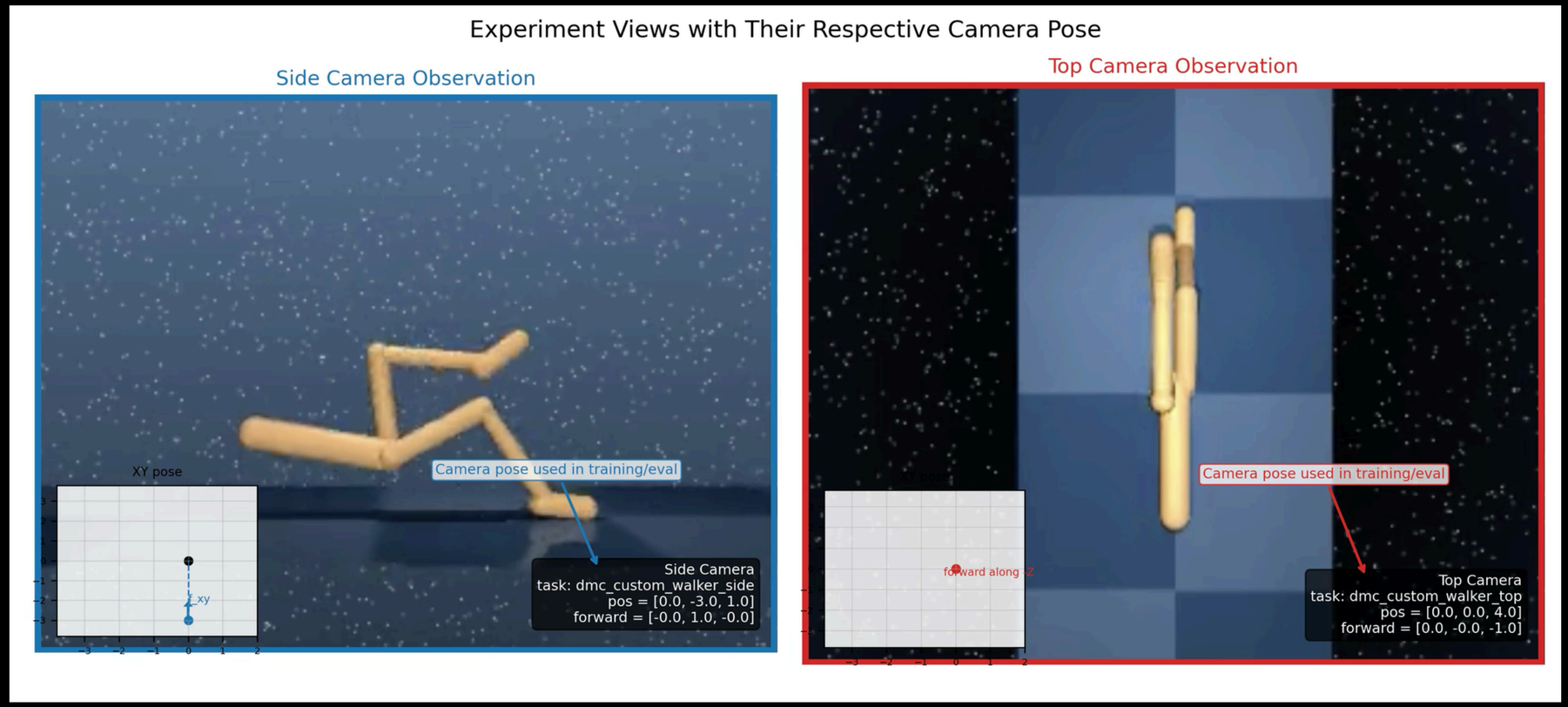
Goal: test how observation viewpoint (camera) affects learning and generalization in a model-based RL agent (DayDreamer).

Hypothesis: training in one viewpoint yields strong in-domain performance but poor transfer to a different viewpoint.

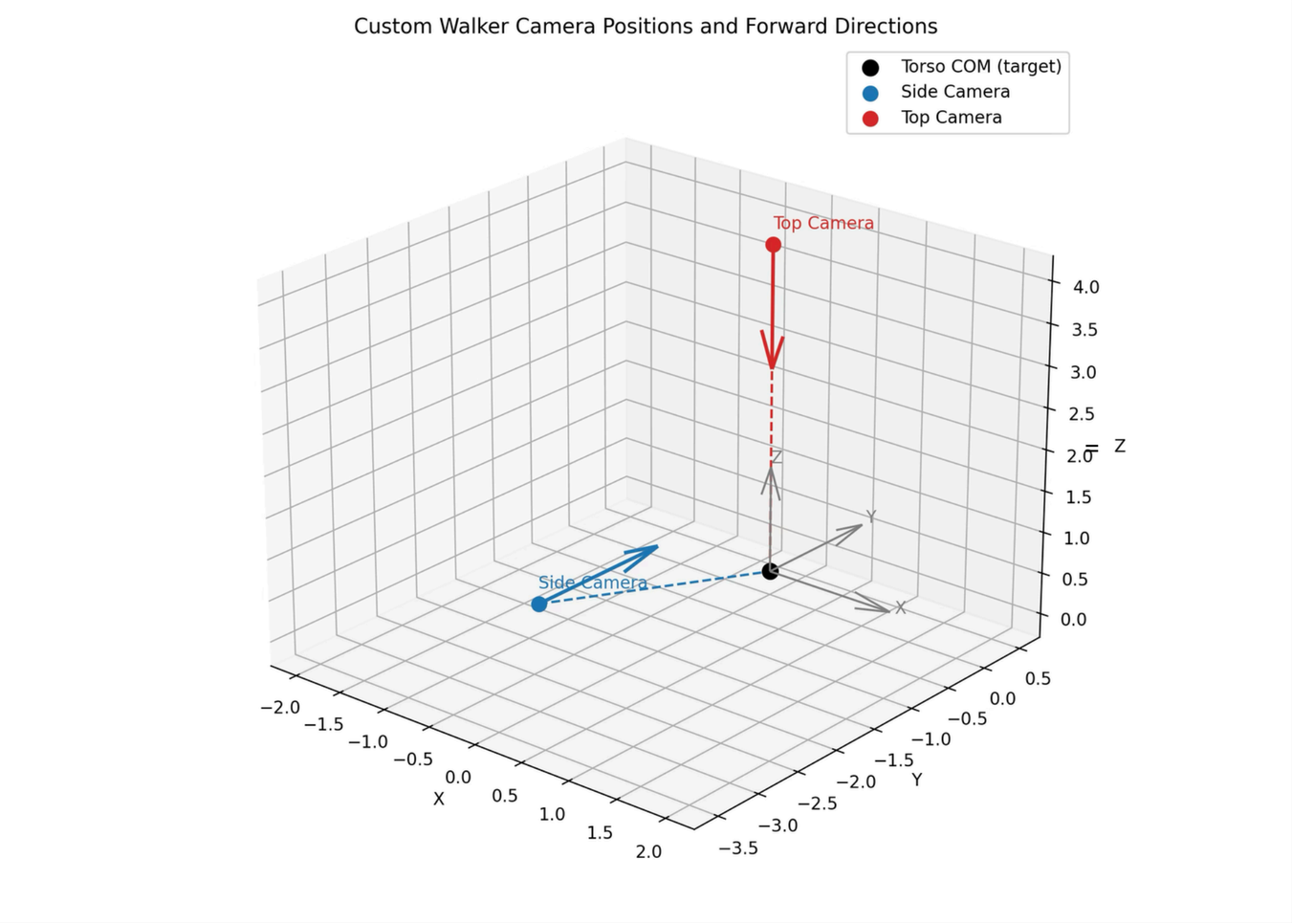
We compare two independently trained agents and evaluate both in-domain and cross-domain.

Experiment Setup (What We Trained and Tested)

Task: `dmc_custom_walker_side` and `dmc_custom_walker_top`.
Same physics task (walker), different camera pose.
Train: 1M environment steps per viewpoint.
Eval: 10 episodes in-domain + 10 episodes cross-domain (transfer).



Camera Pose Used in Experiments



Both cameras use `mode=trackcom`, `target=torso`.
Side: pos [0, -3, 1], forward [0, 1, 0].
Top: pos [0, 0, 4], forward [0, 0, -1].
These define the observation distribution the agent learns from.

Training Setup

Side-view training

```
python -u embodied/agents/dreamerv2plus/train.py \
  --run train \
  --logdir "logdir/run_gruenau_server_side_01" \
  --eval_dir "logdir/run_gruenau_server_side_01/eval_episodes" \
  --task dmc_custom_walker_side \
  --tf.platform gpu \
  --tf.jit True \
  --tf.precision float32 \
  --env.amount 1 \
  --env.parallel none \
  --encoder.cnn_keys '^image$' \
  --decoder.cnn_keys '^image$' \
  --batch_size 16 \
  --train.log_keys_video image \
  --train.steps 1000000 \
  --train.train_every 5 \
  --train.pretrain 100 \
  --train.log_every 100
```

Top-view training

```
python -u embodied/agents/dreamerv2plus/train.py \
  --run train \
  --logdir "logdir/run_gruenau_server_top_01" \
  --eval_dir "logdir/run_gruenau_server_top_01/eval_episodes" \
  --task dmc_custom_walker_top \
  --tf.platform gpu \
  --tf.jit True \
  --tf.precision float32 \
  --env.amount 1 \
  --env.parallel none \
  --encoder.cnn_keys '^image$' \
  --decoder.cnn_keys '^image$' \
  --batch_size 16 \
  --train.log_keys_video image \
  --train.steps 1000000 \
  --train.train_every 5 \
  --train.pretrain 100 \
  --train.log_every 100
```

Learning Performance: Training Returns

Left: raw episode return + 10-episode moving average. Right: running best (solid) and cumulative mean (dashed).

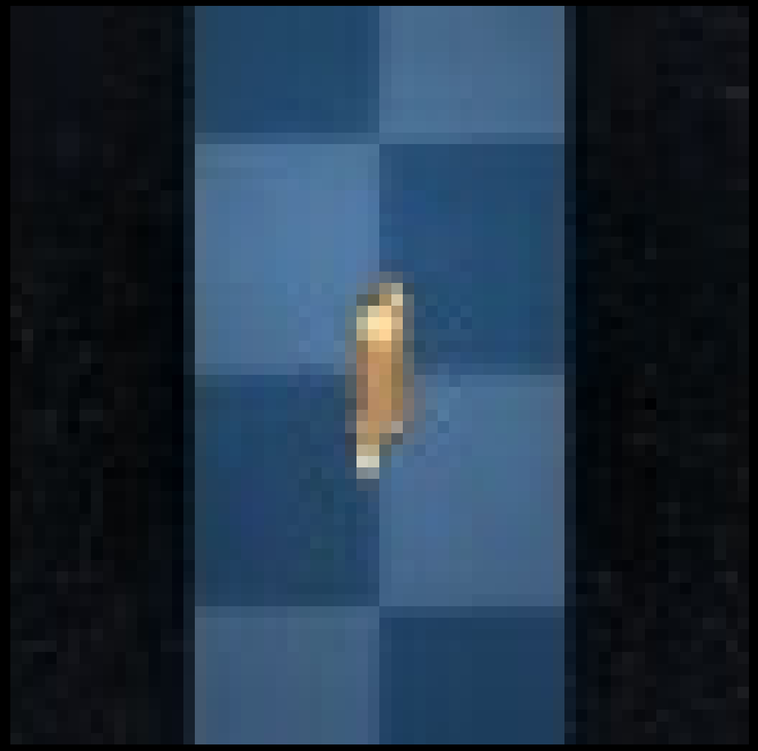
Training Progress (Side View)

step: 0

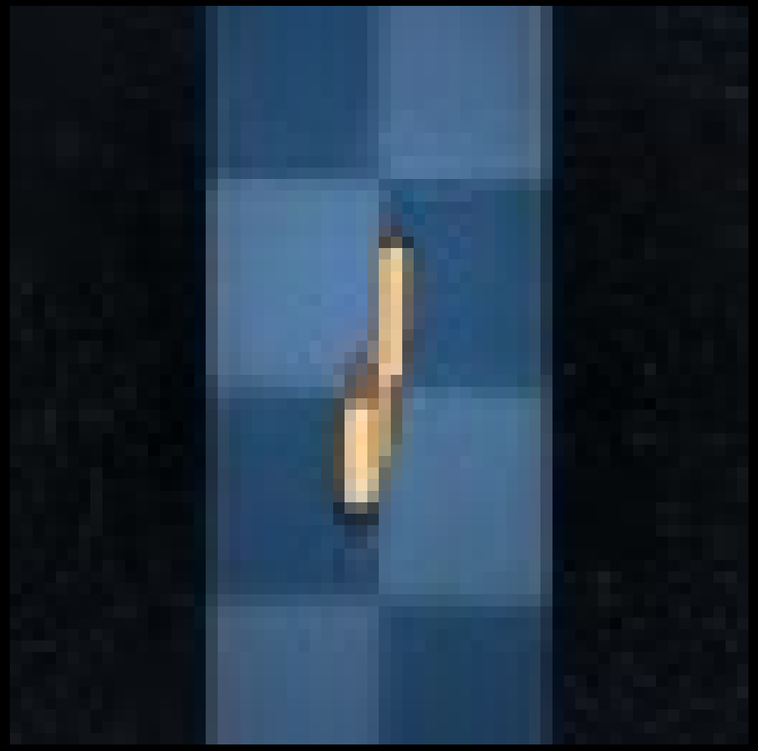
step: 500.000

step: 1.000.000

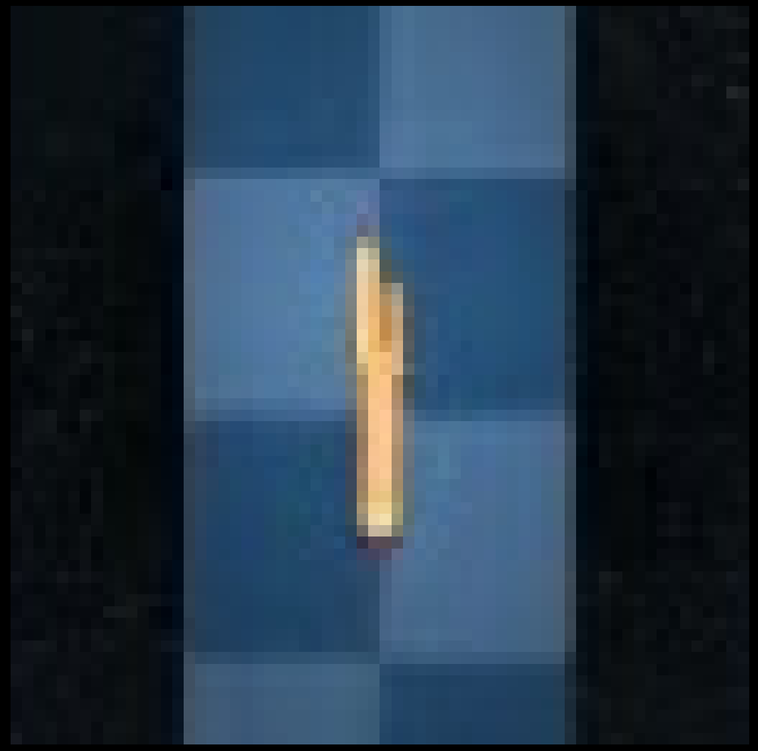
Training Progress (Top View)



step:0



step:500.000



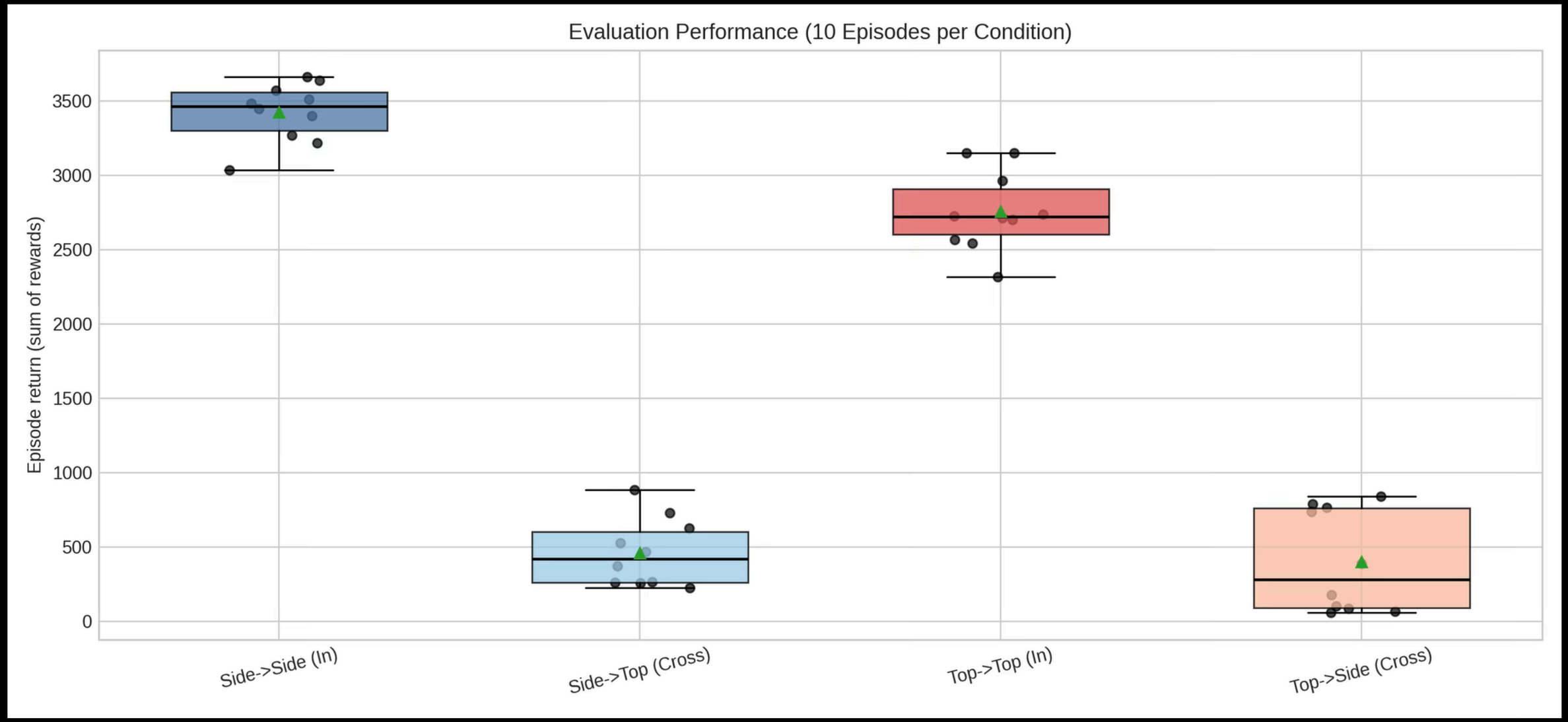
step:1.000.000

Evaluation Conditions (In-Domain vs Cross-Domain)

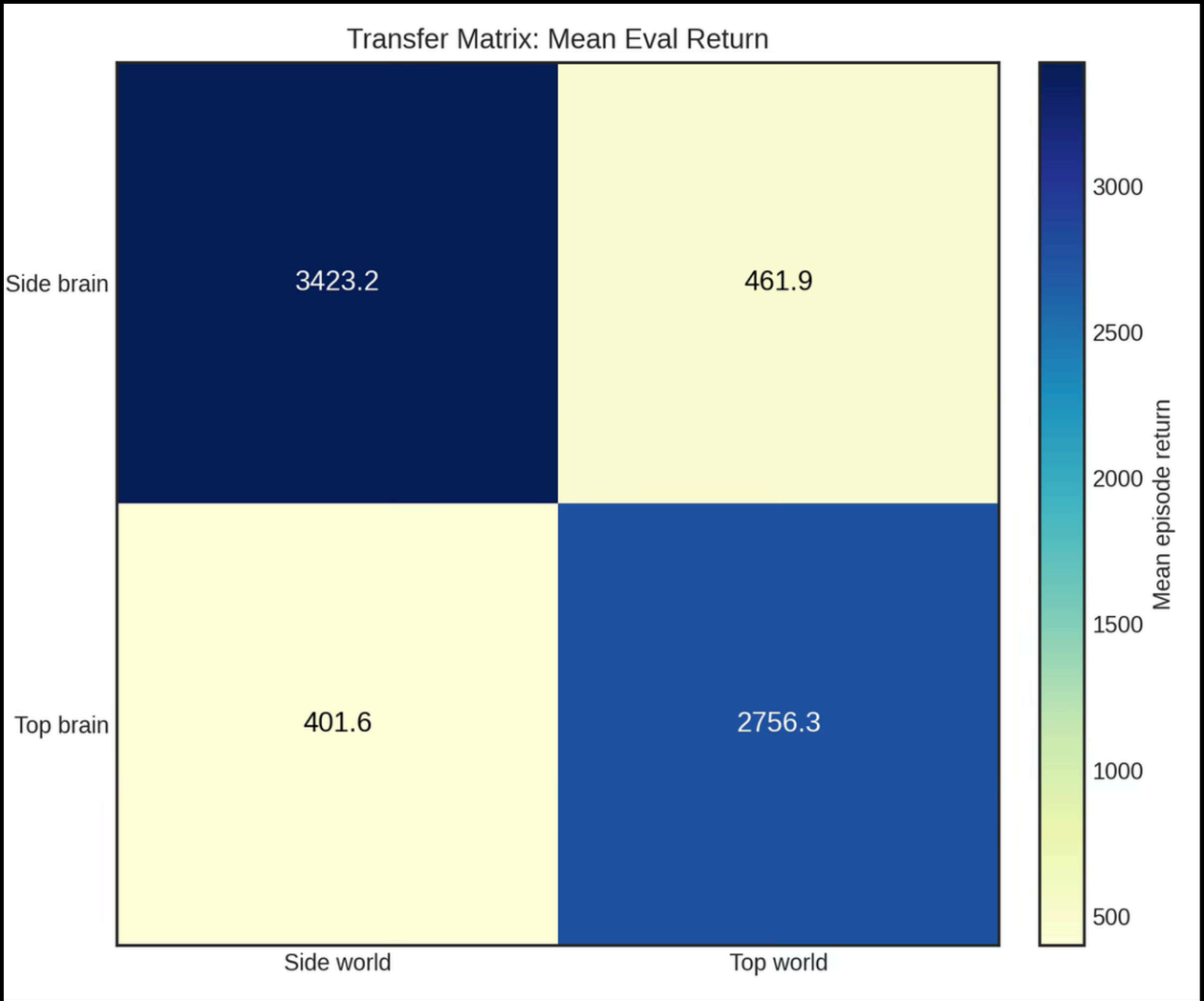
	Side world **(dmc_custom_walker_side)**	**Top world** **(dmc_custom_walker_top)**
Side brain (trained on side)	In-domain Side->Side	Cross-domain Side->Top
Top brain (trained on top)	Cross-domain Top->Side	In-domain Top->Top

- Each condition evaluated with 10 episodes (saved .npz rollouts).
- Score/return = sum of rewards over the episode.
- Cross-domain tests transfer across camera viewpoint.

Evaluation Returns: Distribution Across Conditions

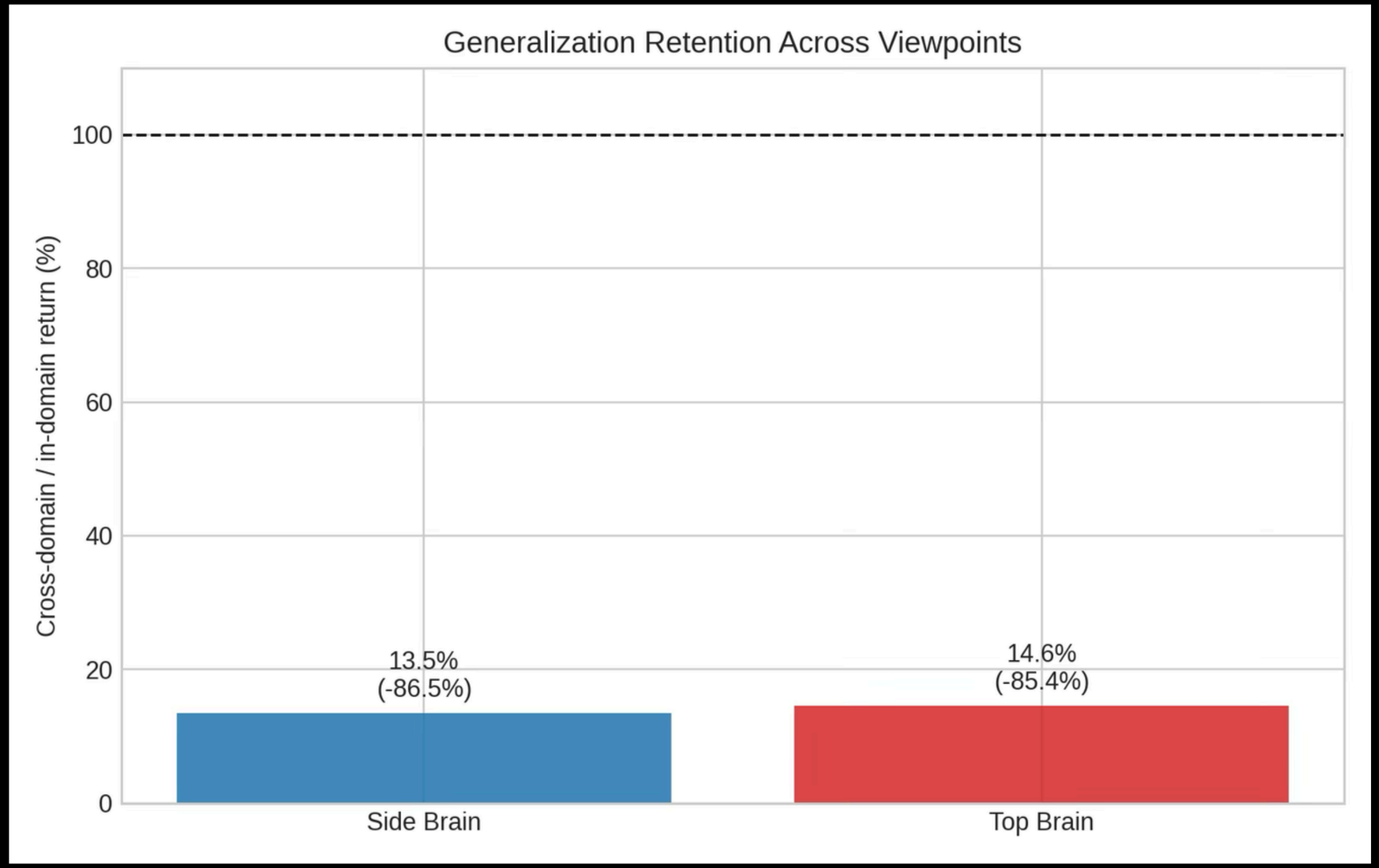


Transfer Matrix (Mean Return)



- Strong diagonal = good in-domain performance.
- Weak off-diagonal = poor viewpoint transfer.
- This suggests learned representations are viewpoint-specific.

Generalization Retention (Cross / In-Domain)

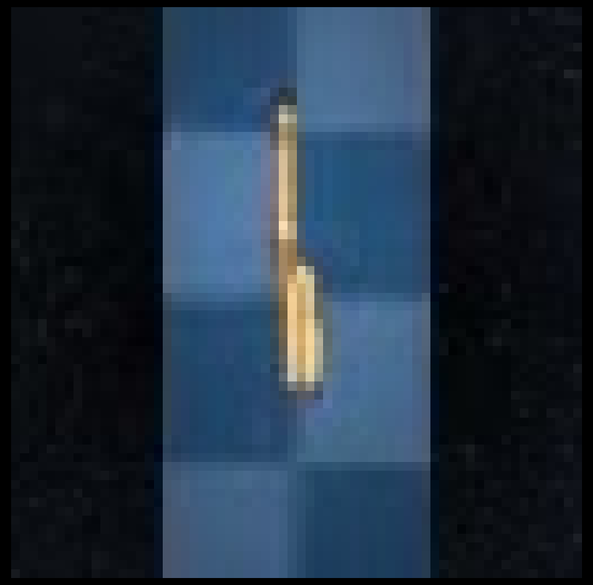


- **Side brain retention: 13.5% (drop 86.5%).**
- **Top brain retention: 14.6% (drop 85.4%).**
- **Takeaway: major performance collapse under pure viewpoint shift.**

In Domain vs Cross Domain Evaluations

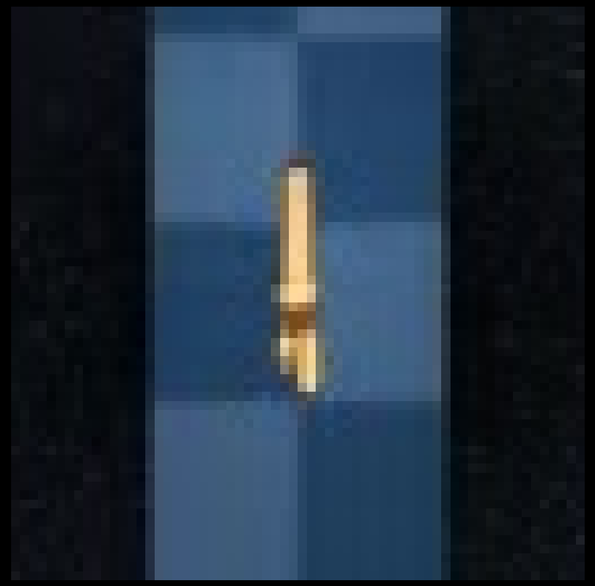
In domain(Sidebrain → Side env.)

In domain(Topbrain -> Top env.)

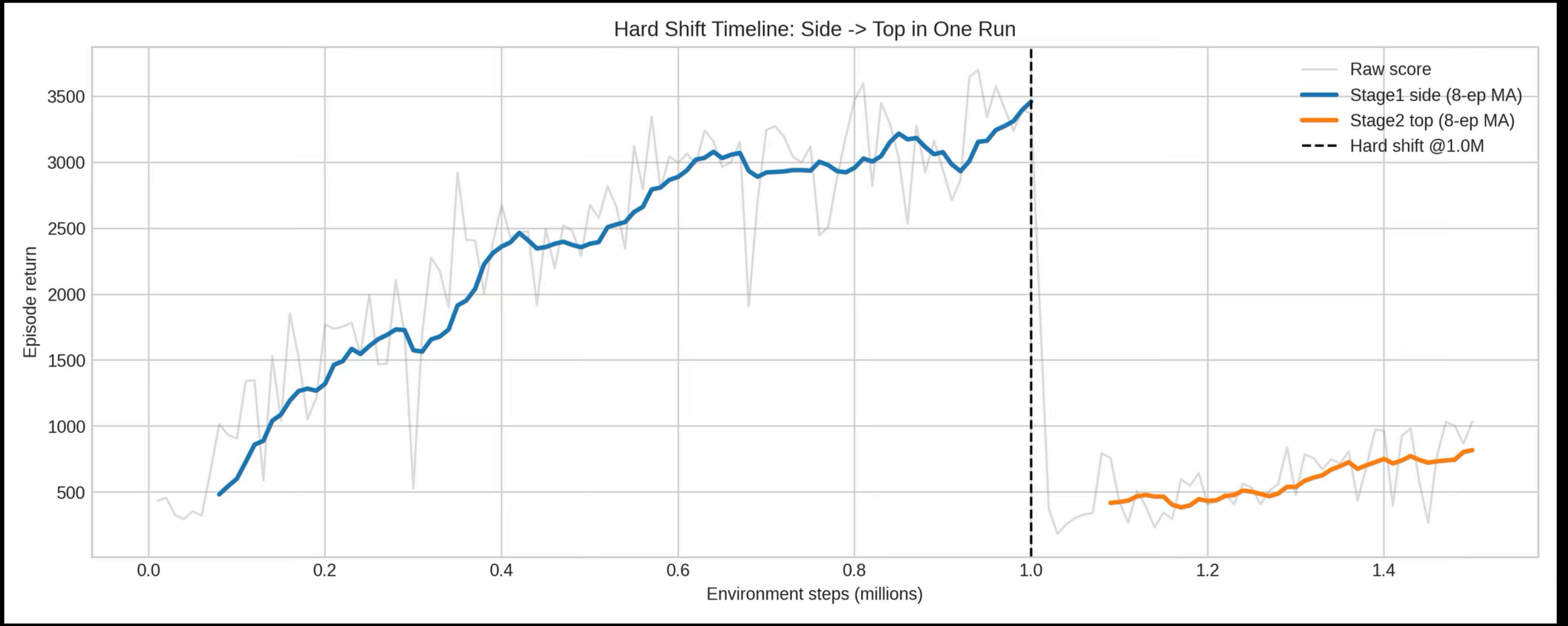


Cross(Topbrain -> Side env.)

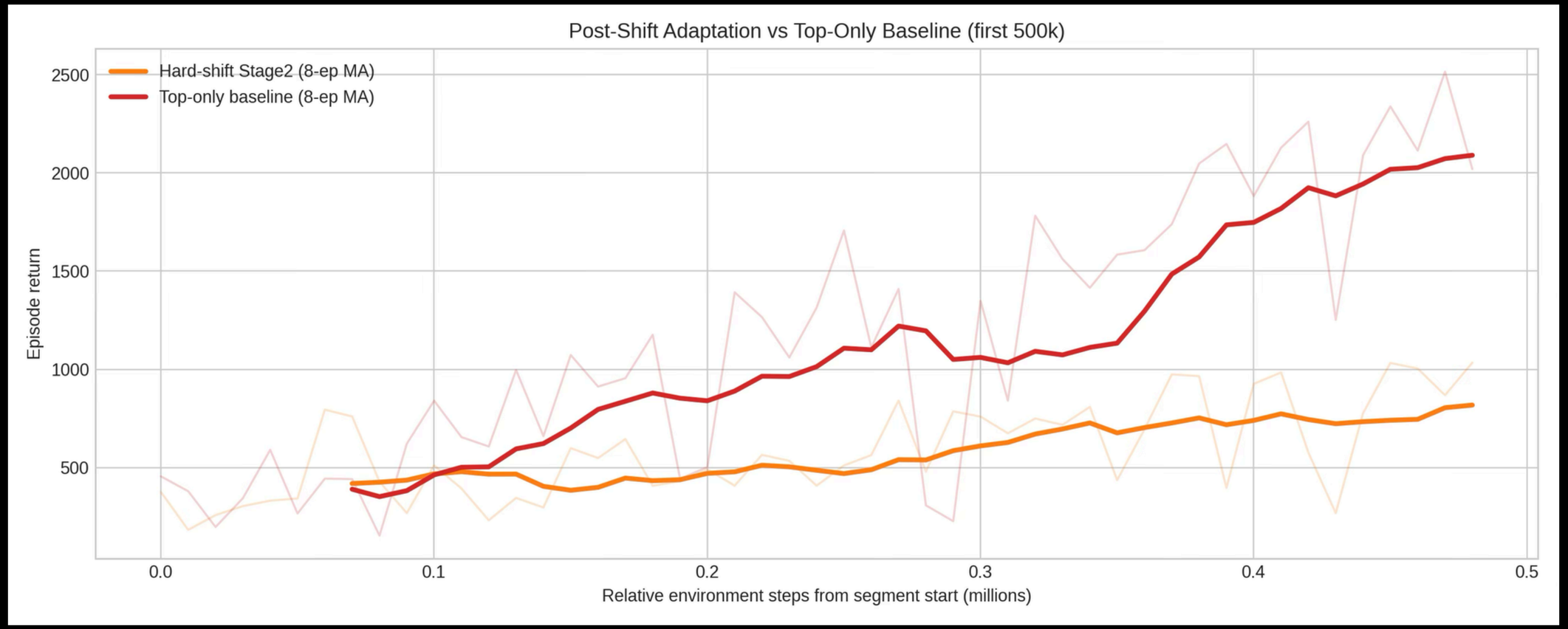
Cross(Sidebrain-> Top env.)



Hard Shift Experiment: Side -> Top in One Run



Post-Shift Adaptation vs Top-Only Baseline (500k)



- **Post-shift first-10 mean: 404 (vs pre-shift last-10: 3327).**
- **Post-shift last-10 mean: 786 (23.6% of pre-shift).**
- **Top baseline last-10 by 500k: 2074.**
- **Hard-shift post last-10 is only 37.9% of top baseline**

Interpretation (What We Learned)

- DayDreamer learns strong policies when train/test perception match.
- Viewpoint shift breaks the observation distribution and likely the learned world model features.
- Result: high specialization, low invariance, and large transfer drop.
- Hard shift confirms catastrophic forgetting / interference under abrupt viewpoint change.
- Improvement ideas: multi-view training, camera randomization, replay balancing, stronger exploration.

Limitations & Future Work

Limitations

- Results are based on one training run per condition (no multi-seed significance).
- Only one type of domain shift was studied (camera viewpoint).
- Evaluation uses 10 episodes per condition, sufficient for trends but limited for strict statistical claims.

Future Work

- Run multiple seeds to quantify variance and robustness.
- Study additional perceptual shifts (camera position, noise, occlusion).
- Train with multi-view or augmented observations to encourage viewpoint-invariant representations(camera randomization).
- Analyze latent state structure to better understand representation specialization.

Conclusion

- DayDreamer learns strong, data-efficient locomotion policies when training and evaluation perception match.
- Performance is strongly viewpoint-dependent: side-view training outperforms top-view training on this task.
- Cross-view generalization is poor for both models, with large performance drops under camera changes.
- Perception critically shapes the learned world model: representations are specialized to the training viewpoint.
- Key takeaway: model-based RL learns what it sees and limited perceptual diversity can severely limit transfer.

THANK YOU.

Do you have any
questions?