Мультимодальные и поведенческие модели

Offline Reinforcement Learning and Human-in-the-Loop

MIPT VLA 2026 | 16 ноября 2026

Алексей Староверов



МФТИ

Center for Cognitive Modeling

План лекции

Task: LeHome challenge 2026

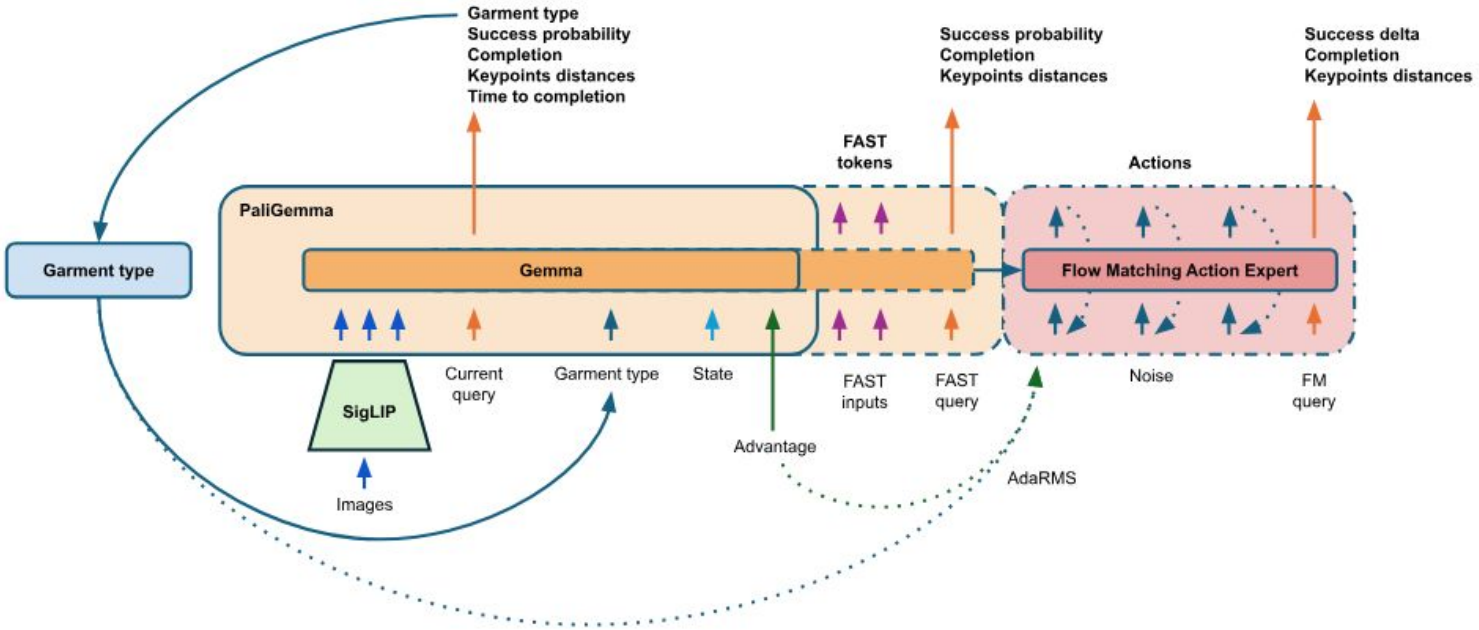
Offline Reinforcement Learning

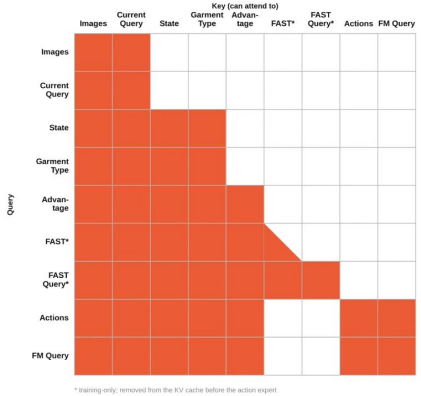
Sim2Real

LeHome challenge 2026



Model Arcitechture





auxiliary prediction heads that make the policy partly its own value function: a single query token feeds cheap linear heads that, from the same forward pass, predict success probability, task completion, garment type, and — 30 frames ahead — the keypoint distances that define success plus a Q-function residual;

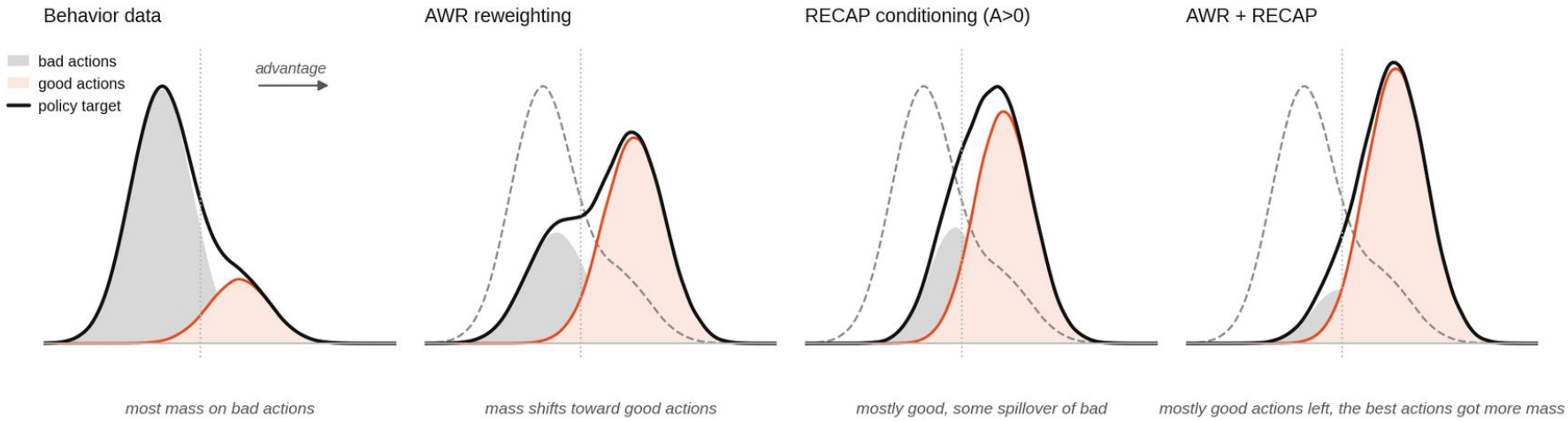
a **garment-type input token**, with the type inferred at the start of each episode since it is hidden at evaluation;

advantage conditioning (the RECAP style) plus multi-signal AdaRMS conditioning, so the advantage and garment signals reach every layer of the action expert;

exclusive self-attention across the VLM and the action expert;

keypoint-distance and future-prediction heads as a very cheap world-model substitute — predicting the few numbers that matter instead of whole future frames.

A reinforcement learning loop



AWR weights the training data toward high-advantage frames — applied through sampling rather than the loss, so good frames are simply loaded more often.

RECAP feeds the advantage in as a conditioning input, in effect telling the model to "predict good actions only" — which also unlocks classifier-free guidance (CFG) at inference.

AWR: Advantage-Weighted Regression

Less on-policy methods use data collected by the “older” policy:

$$L(\theta) = \mathbb{E}_{s,a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} \cdot A(s,a)\right]$$

One of the most popular methods - PPO, adds extra tricks:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\,\hat{A}_t,\ \text{clip}\big(r_t(\theta),\,1-\varepsilon,\,1+\varepsilon\big)\,\hat{A}_t\big)\Big]$$

AWR: Advantage-Weighted Regression

Off-policy/Offline - Advantage-Weighted Regression (AWR)

$$\max_{\pi} \; \mathbb{E}_{s,a\sim\mu}\left[\frac{\pi(a \mid s)}{\mu(a \mid s)} A^{\mu}(s,a)\right] \qquad \text{s.t.} \quad D_{\mathrm{KL}}\big(\pi \,\|\, \mu\big) \le \varepsilon$$

Closed form solution

$$\pi^{*}(a \mid s) \;\propto\; \mu(a \mid s) \cdot \exp\left(\frac{A^{\mu}(s,a)}{\beta}\right)$$

$$L(\theta) \;=\; \mathbb{E}_{s,a\sim\mu,\,\tau,\,\varepsilon}\Big[\exp\big(A(s,a)/\beta\big) \cdot \big\|v_{\theta}(x^{\tau},s) - (\varepsilon - a)\big\|^{2}\Big]$$

AWR: Advantage-Weighted Regression

I use β=1 and clip A in [-2,2]:

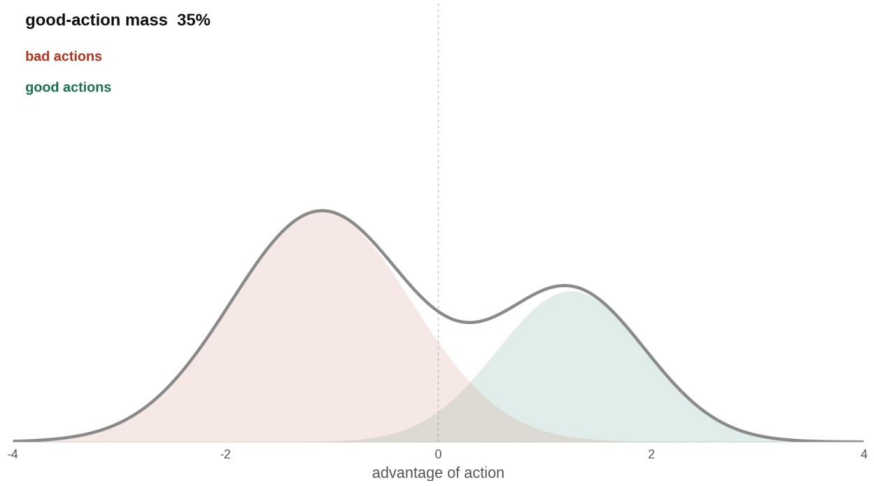
$$L(\theta) = \mathbb{E}_{s,a\sim\mu,\,\tau,\,\varepsilon}\Big[\, e^{\,\mathrm{clip}(A_i,\,-2,\,2)} \cdot \big\| v_\theta(x^\tau, s) - (\varepsilon - a) \big\|^2 \Big]$$

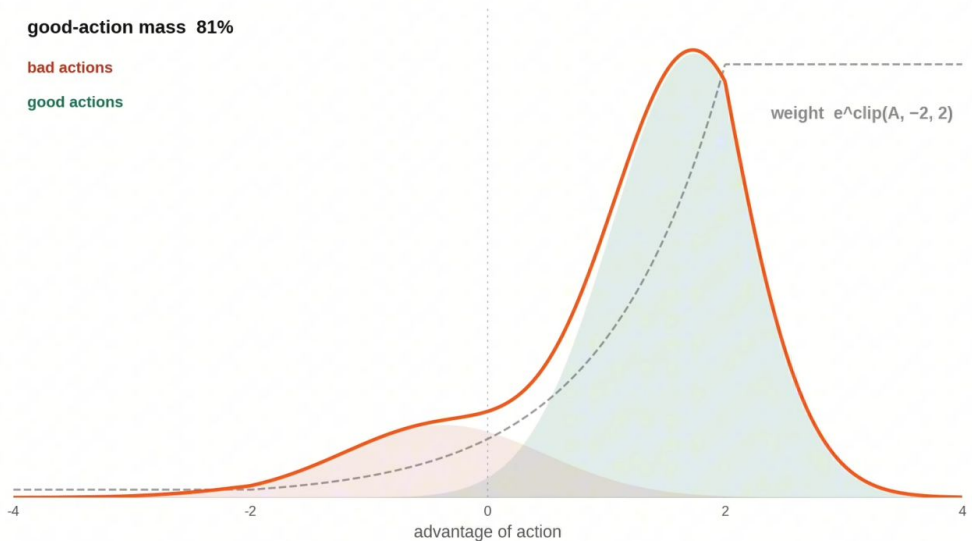
And I replaced loss weights with sampling probability:

$$L(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D},\,\tau,\,\varepsilon}\Big[\, \big\| v_\theta(x^\tau, s) - (\varepsilon - a) \big\|^2 \Big]$$

$$P(\text{sample frame } i) \;\propto\; e^{\,\mathrm{clip}(A_i,\,-2,\,2)}$$

AWR: Advantage-Weighted Regression





RECAP intuition

$$\mu(a \mid s) = \underbrace{\mu(a \mid s,\ A > \varepsilon)}_{\text{the good slice}} \cdot \underbrace{P(A > \varepsilon \mid s)}_{\text{share of good actions}} + \underbrace{\mu(a \mid s,\ A \le \varepsilon) \cdot P(A \le \varepsilon \mid s)}_{\text{everything else}}$$

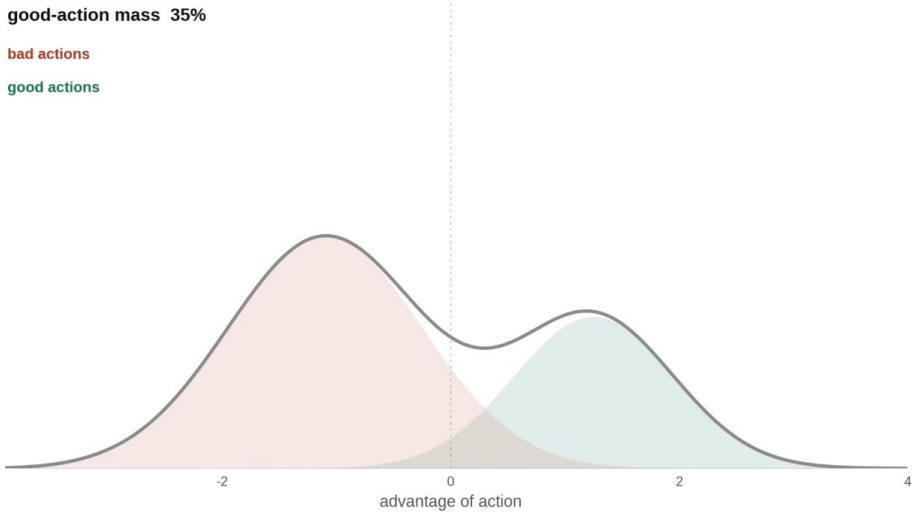
Unconditional path:

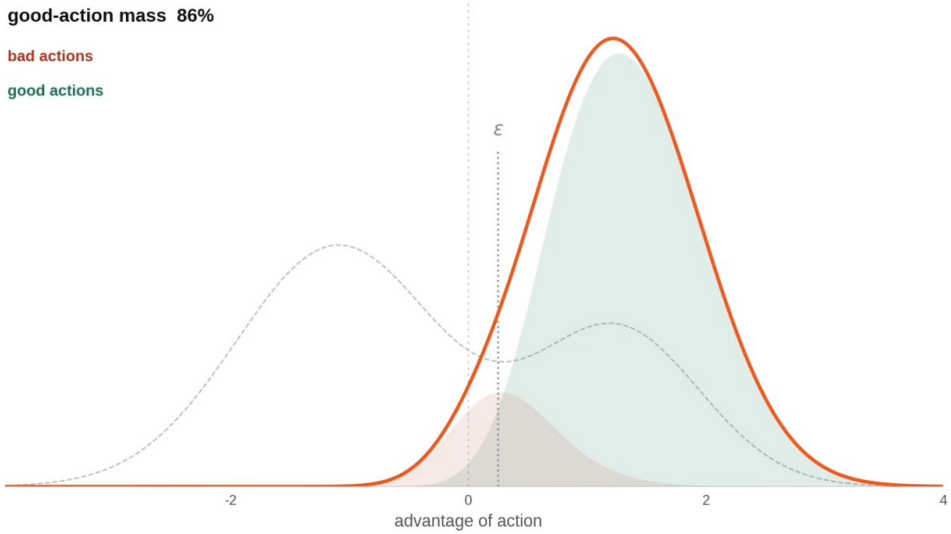
$$\pi_\theta(a \mid s,\ \varnothing) \ \rightarrow\ \mu(a \mid s)$$

Conditional “good actions” path:

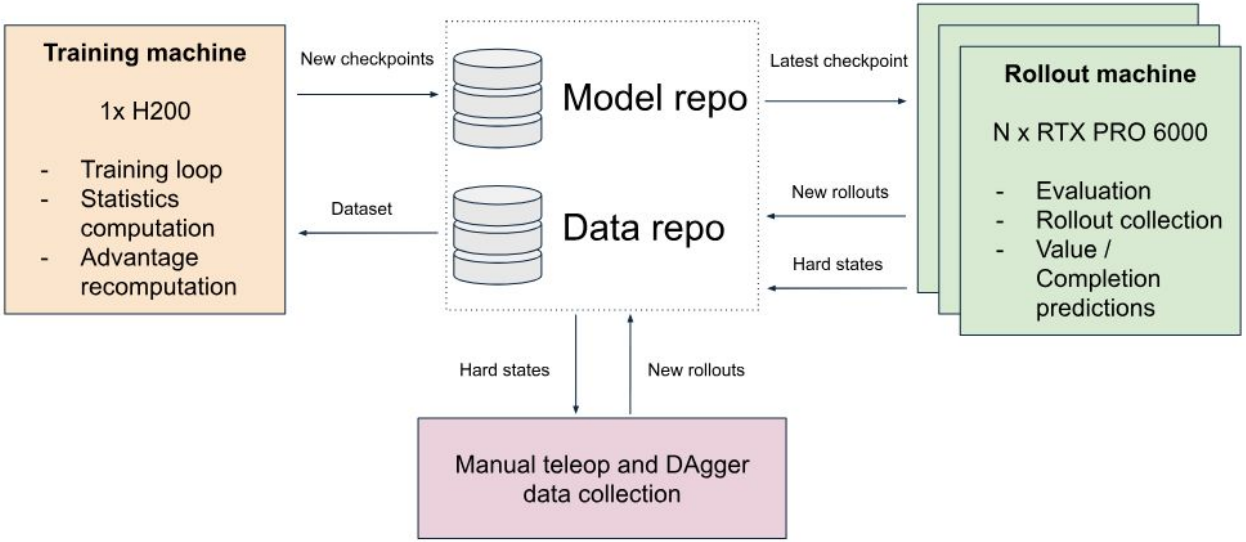
$$\pi_\theta(a \mid s,\ o{=}1) \ \rightarrow\ \mu\big(a \mid s,\ A > \varepsilon\big)$$

RECAP intuition





The asynchronous RL loop

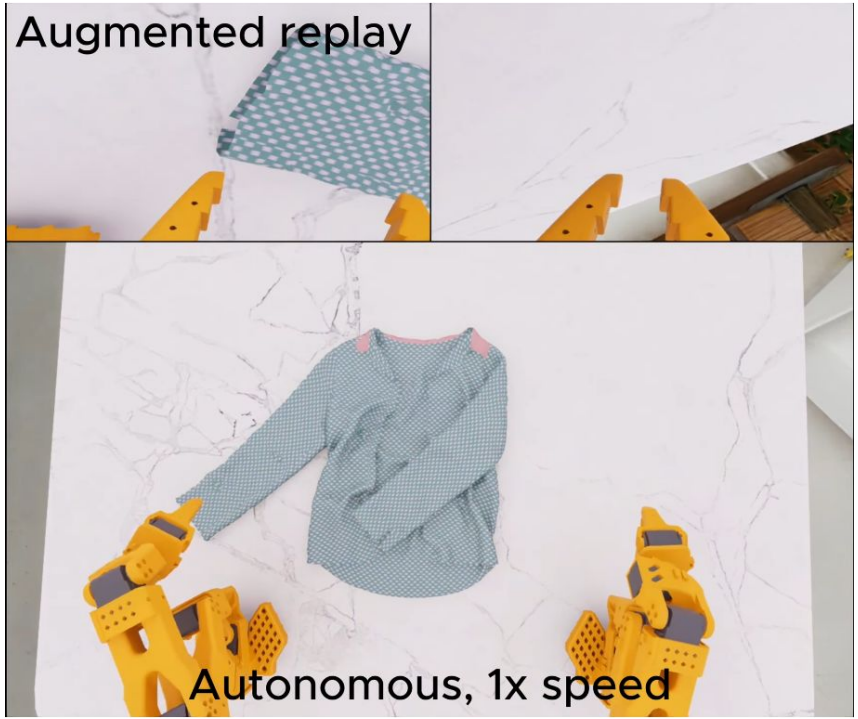


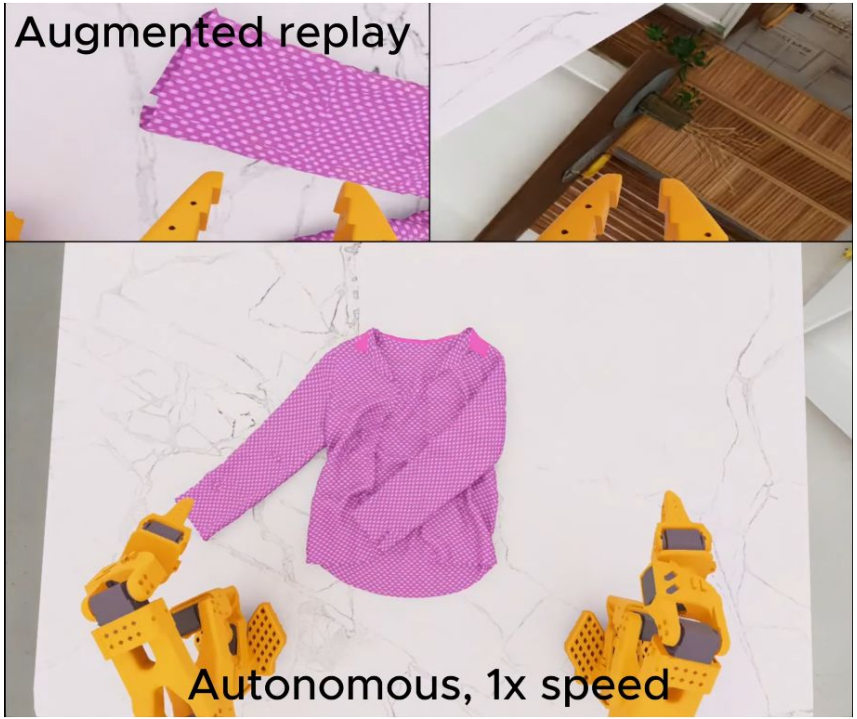
One training machine (1× H200) trains continuously on all available data, recomputing advantages across every rollout dataset before each iteration and shipping a fresh checkpoint roughly every 40 minutes.

One or more rollout machines pull the latest checkpoint and collect as many new rollouts as possible, using several strategies — random, curriculum, success-replay, hard-mining.

When needed, a human operator collects more data by hand — either fresh teleop rollouts or automatically-detected hard cases, for efficient DAgger-style correction.

Rollout collection and optimizations (augmentation + SR based sampling)



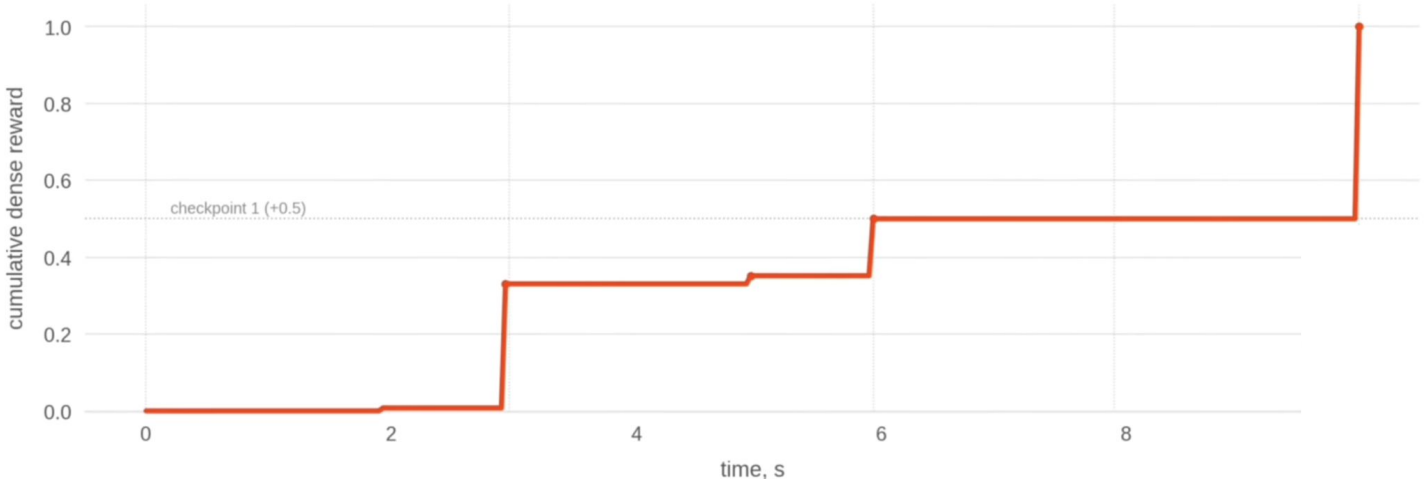


Reward Engineering

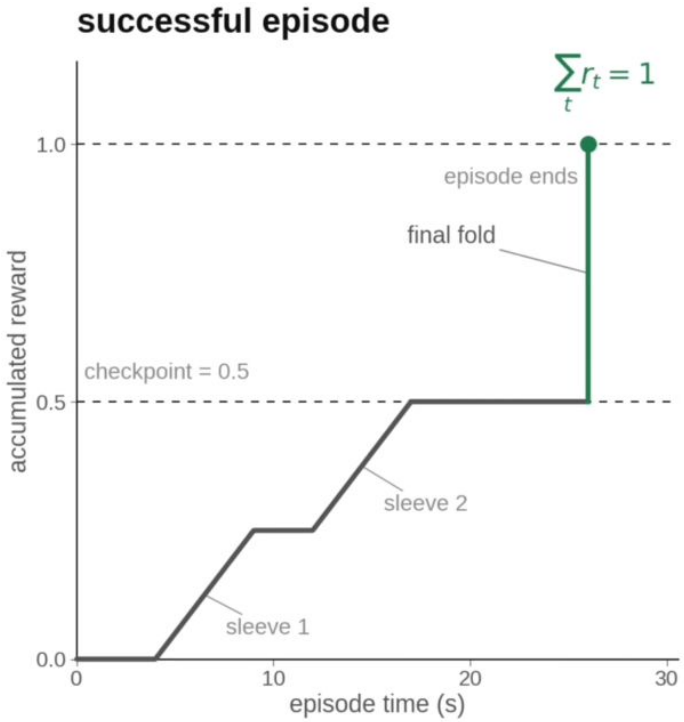
1. checkpoints from the environment: r_t^{cp} (a subset of the success conditions)
2. change of the value prediction: $\delta_t = r_t + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t)$
3. completion progress: $\delta_t^c = \alpha_c\left(\gamma \hat{C}_{t+1} - \hat{C}_t\right)$
4. relative success (GRPO-style): $A_i^{\text{seg}} = \sum_t r_t^{(i)} - \frac{1}{N_g} \sum_{j \in g} \sum_t r_t^{(j)}$

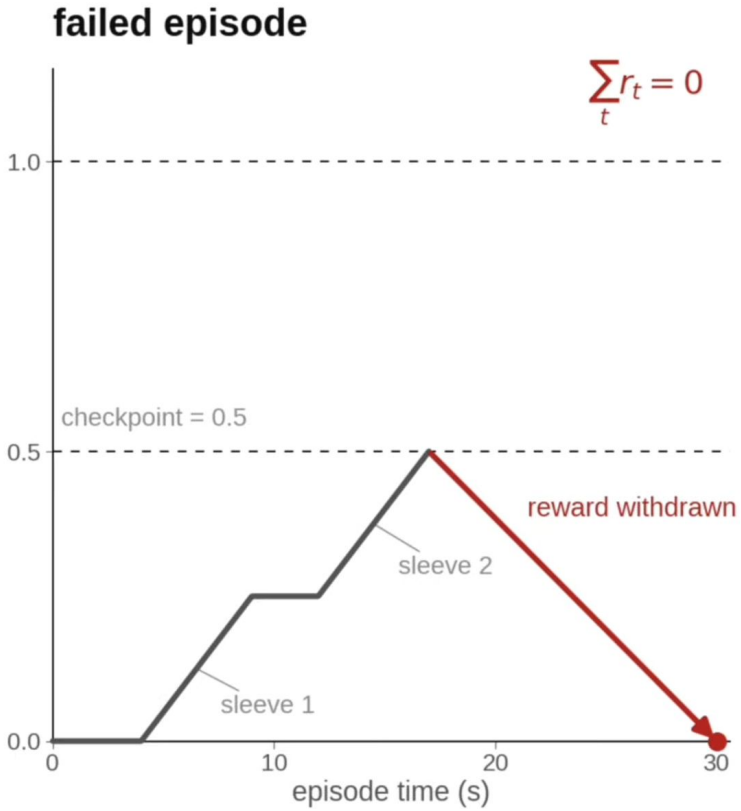
Reward Engineering





Reward Engineering





Reward Engineering

$$G_t = \sum_{k \geq 0} r_{t+k} \qquad V(s) = \mathbb{E}[G_t \mid s_t = s]$$

Monte-Carlo advantage estimation:

$$\hat{A}_t = G_t - V(s_t),$$

$$\mathbb{E}\left[\hat{A}_t \mid s_t, a_t\right] = Q(s_t, a_t) - V(s_t) = A(s_t, a_t)$$

Reward Engineering

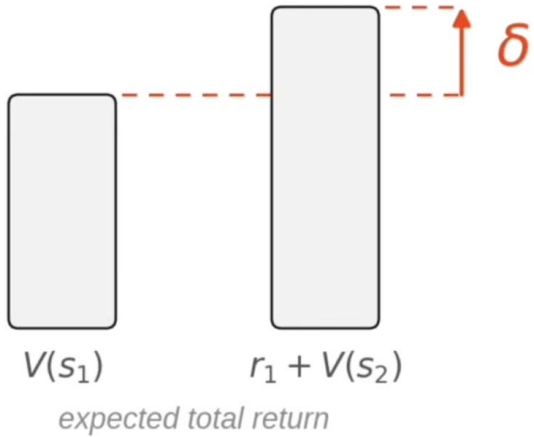
$$\mathrm{Var}(G_t) = \underbrace{\mathrm{Var}\big(V(s_t)\big)}_{\text{state luck}} + \underbrace{\mathbb{E}\big[\mathrm{Var}(G_t \mid s_t)\big]}_{\text{action choice + outcome}}$$

$$\mathrm{Var}\big(G_t - V(s_t)\big) = \mathbb{E}\big[\mathrm{Var}(G_t \mid s_t)\big]$$

$$V(s) = \mathbb{E}[G_t \mid s_t = s] = \arg\min_{b(s)} \mathrm{Var}\big(G_t - b(s_t)\big)$$

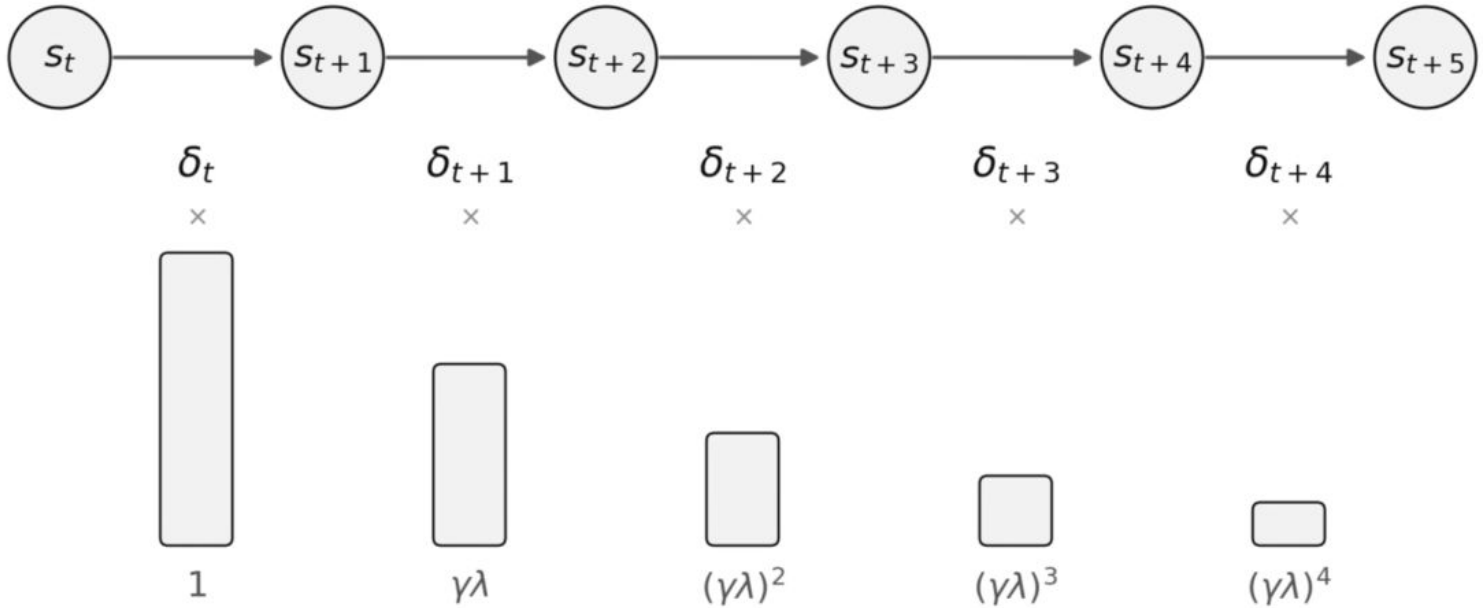
Reward Engineering





$$\delta = r_1 + V(s_2) - V(s_1)$$

Reward Engineering



$$\hat{A}_t = \sum_{k \geq 0} (\gamma\lambda)^k \delta_{t+k}$$

Reward Engineering

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \qquad \hat{A}_t^{\mathrm{GAE}} = \sum_{k \geq 0} (\gamma\lambda)^k \, \delta_{t+k}$$

$\lambda = 0$	$\hat{A}_t = \delta_t$	one-step TD: low variance, biased
$\lambda = 1$	$\hat{A}_t = G_t - V(s_t)$	Monte-Carlo: unbiased, noisy
$\lambda = 0.99, \ \gamma = 0.999$		almost Monte-Carlo, slightly smoothed

Reward Engineering

$$G_t^{\text{remaining}} = \begin{cases} 1 - R_t^{\text{cum}} & \text{success} \\ -R_t^{\text{cum}} & \text{failure} \end{cases} \qquad (R_2^{\text{cum}} - R_1^{\text{cum}} = r_1)$$

$$V(s_t) = P \cdot \left(1 - R_t^{\text{cum}}\right) + (1 - P) \cdot \left(-R_t^{\text{cum}}\right) = P\left(\text{success} \mid s_t\right) - R_t^{\text{cum}}$$

$$\begin{aligned} \delta &= r_1 + V(s_2) - V(s_1) \\ &= r_1 + P\left(\text{success} \mid s_2\right) - R_2^{\text{cum}} - P\left(\text{success} \mid s_1\right) + R_1^{\text{cum}} \\ &= P\left(\text{success} \mid s_2\right) - P\left(\text{success} \mid s_1\right) \end{aligned}$$

Reward Engineering

$$V(s) \;=\; \mathbb{E}[G_t \mid s_t = s] \;=\; \arg\min_{b(s)} \; \mathrm{Var}\big(G_t - b(s_t)\big)$$

$$\hat{V}(s) \;=\; V(s) + \varepsilon$$

$$\mathrm{Var}\big(G - \alpha\hat{V}\big) = \mathrm{Var}(G) - 2\alpha\,\mathrm{Cov}(G, \hat{V}) + \alpha^2\,\mathrm{Var}(\hat{V}) \quad \Rightarrow \quad \alpha^* = \frac{\mathrm{Cov}(G, \hat{V}}{\mathrm{Var}(\hat{V})}$$

$$\hat{A} \;=\; G - \alpha_s\,\hat{V}(s)$$

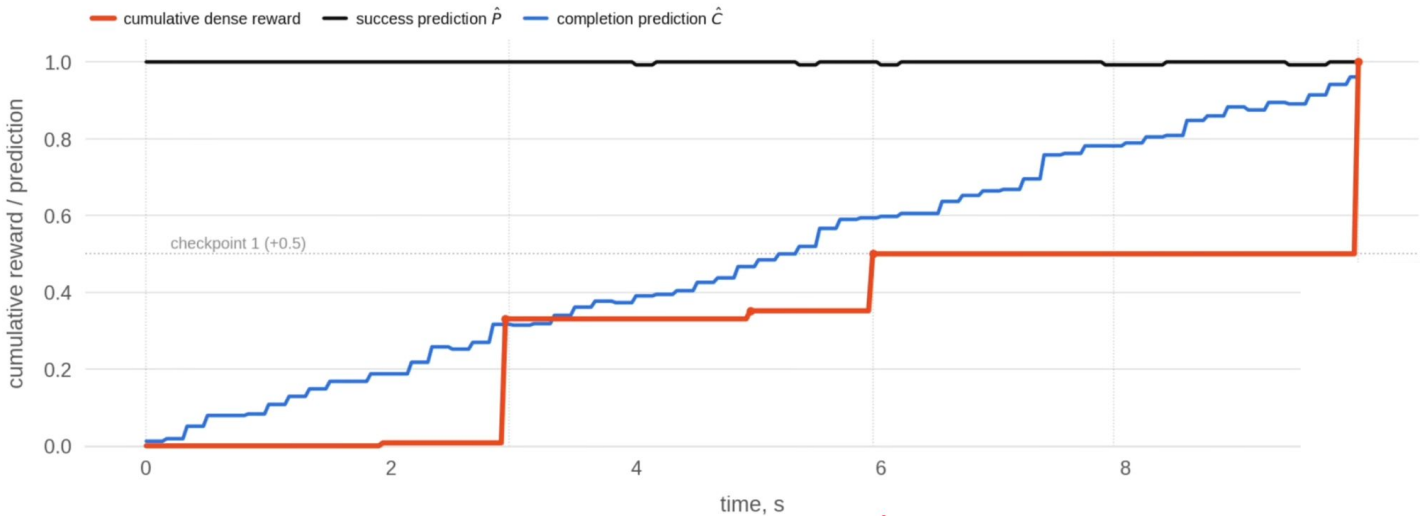
Reward Engineering

$$\hat{A} = G - \alpha_s \hat{V}(s), \qquad \delta = r_1 (1 - \alpha_s) + \alpha_s (\hat{P}_2 - \hat{P}_1)$$

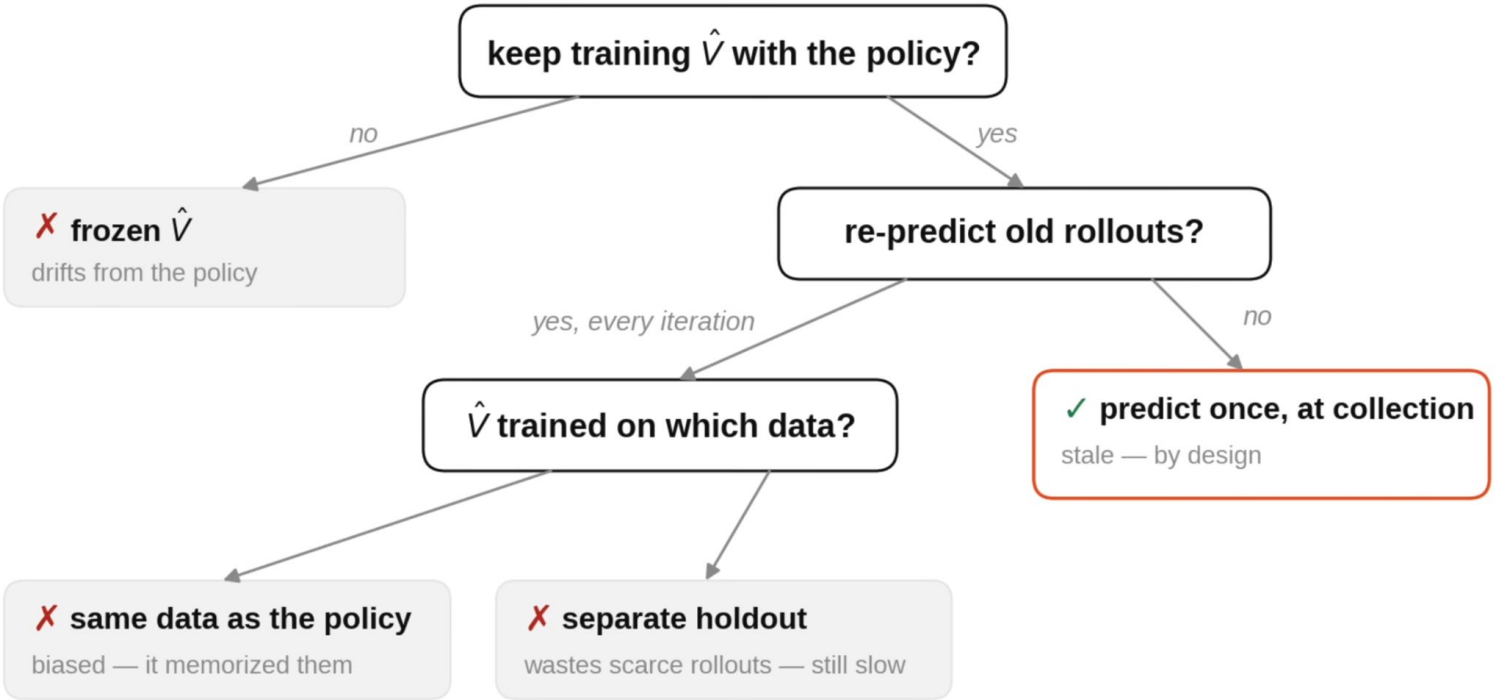
$\alpha_s = 0$	$\Rightarrow$	$\delta = r_1$	pure checkpoint reward (prediction useless)
$\alpha_s = 1$	$\Rightarrow$	$\delta = \hat{P}_2 - \hat{P}_1$	pure probability change (prediction perfect)
$\alpha_s = 0.5$	$\Rightarrow$	both	imperfect prediction — both signals stay alive

Reward Engineering





Reward Engineering

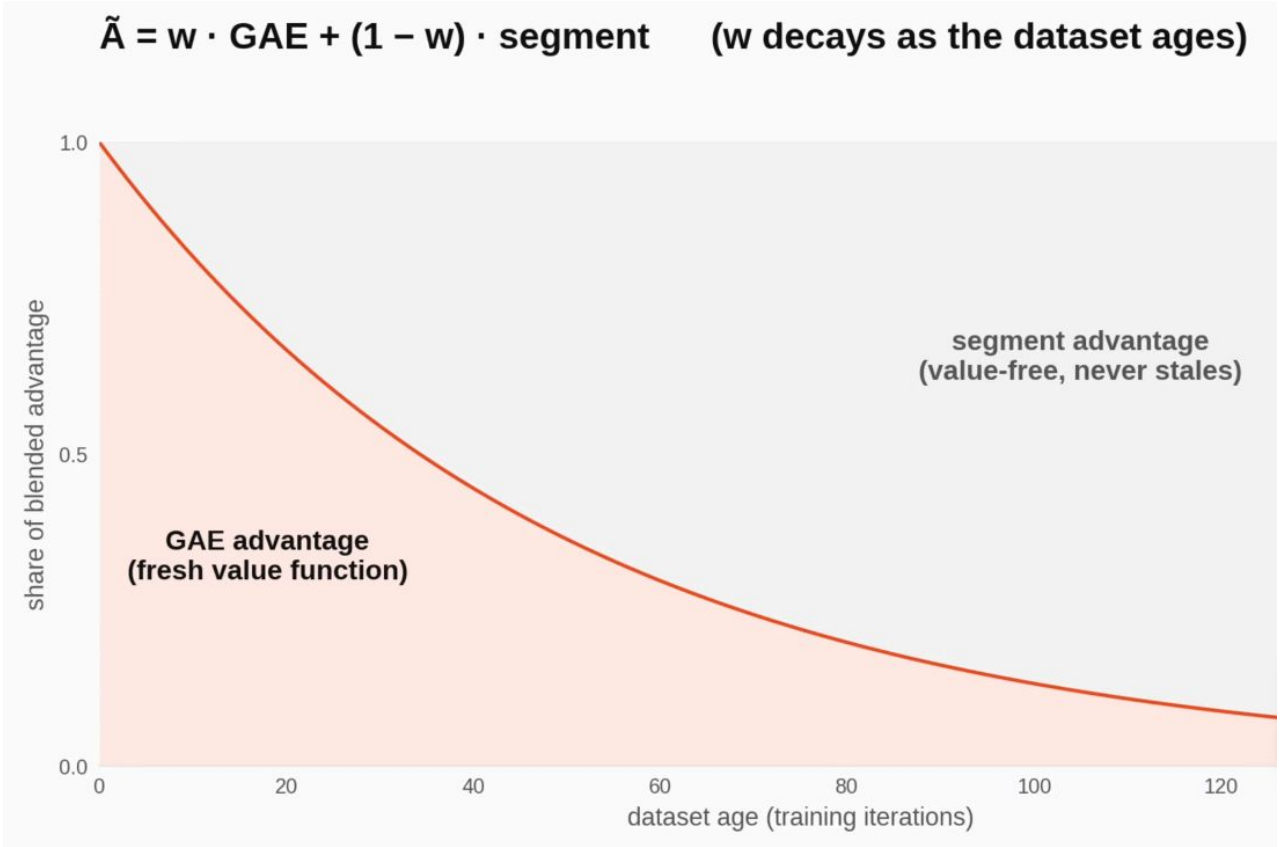


Reward Engineering

Relative success — GRPO with garment groups

$$A_i^{\text{seg}} = \sum_t r_t^{(i)} - \frac{1}{N_g} \sum_{j \in g} \sum_t r_t^{(j)}, \qquad g = \text{episodes with the same garment}$$

Reward Engineering



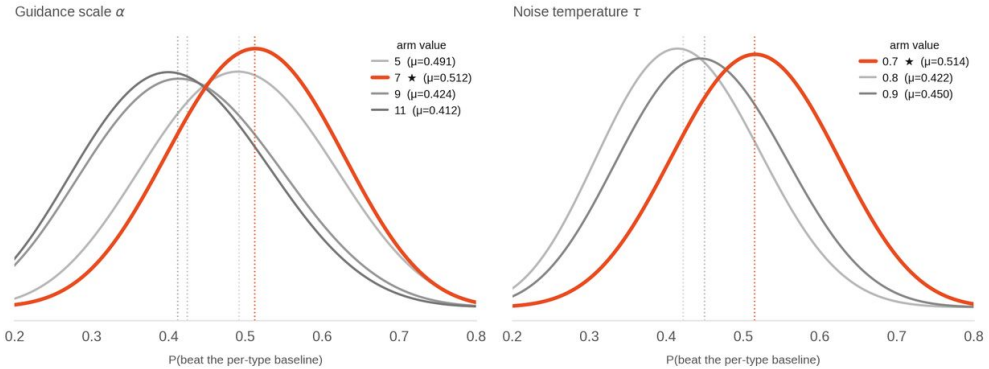
Reward Engineering

$$\hat{A}_t = w \cdot \underbrace{A_t^s}_{\text{Checkpoints + Success Probability}} + (1-w) \cdot \underbrace{A_t^{\text{seg}}}_{\text{Relative success}} + \underbrace{\Phi_t^c}_{\text{Completion}}$$

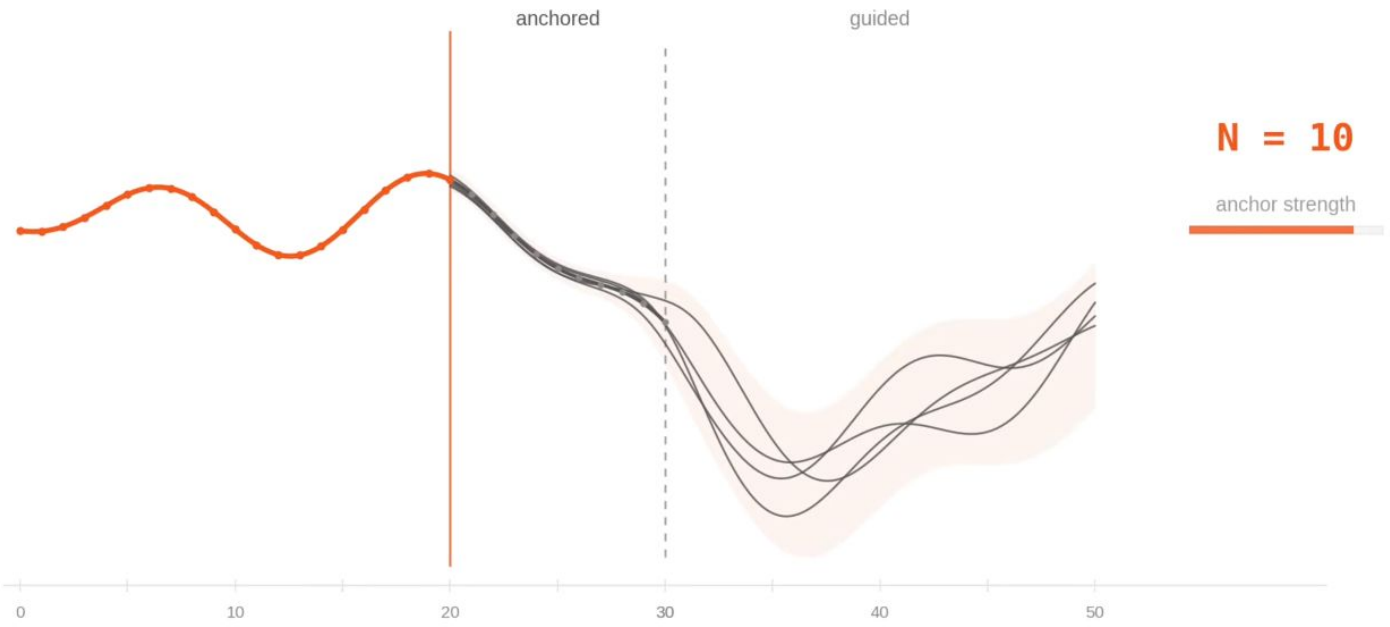
- **objective task progress** from the challenge's own keypoint checkpoints (with all reward withdrawn on failure, so the episode return stays binary);
- **success-probability predictions**, used as a value baseline;
- **completion-percentage predictions**, a progress signal that stays stable as the policy evolves;
- **relative success across garments**, which stale rollouts fall back on as their own predictions go out of date.

Inference-time optimization (Thompson sampling)

- **Execution length** — how many of the 30 predicted actions to run before re-planning against a fresh observation.
- **Playback speed** — a time-stretch of the executed actions, so the arms move faster or slower.
- **Inpainting overlap** — how long the tail of the previous chunk softly anchors the next one before it is freed to self-correct.
- **Guidance scale** — how hard classifier-free guidance amplifies the advantage conditioning. It converged surprisingly high (7–9).
- **Noise temperature** — scales the initial flow-matching noise to concentrate candidates nearer the distribution mode.
- **Best-of-N** — draw N candidate chunks from the same prefix and execute the one the Q-head scores highest.

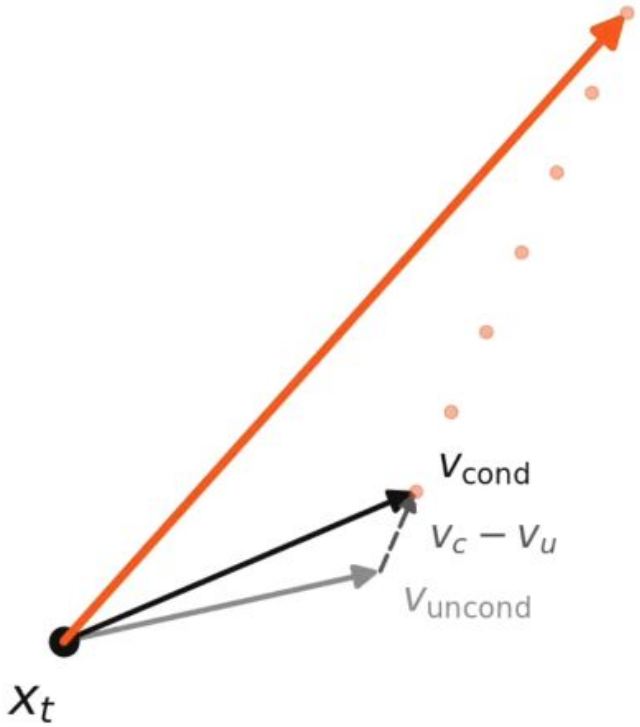


Inference-time optimization (inpainting overlap)



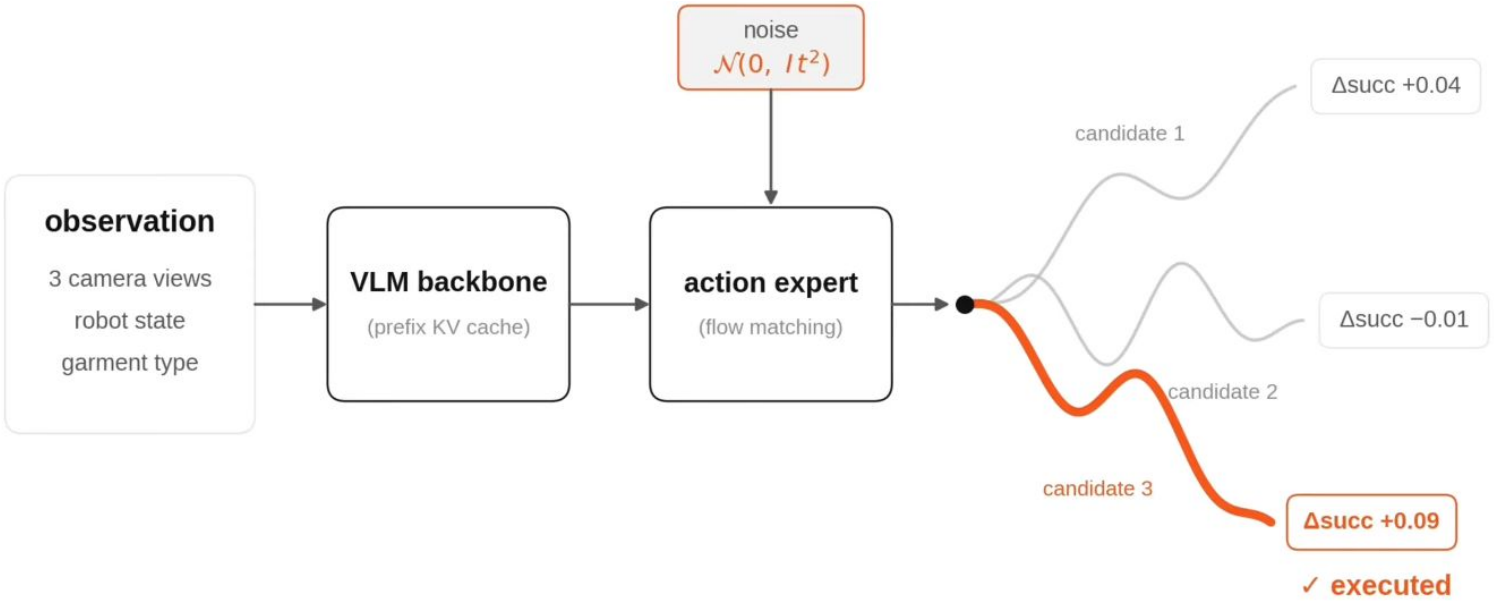
Inference-time optimization (CFG - classifier-free guidance)

$$\hat{v} = v_u + \alpha\,(v_c - v_u)$$



$\alpha = 7.0$

Inference-time optimization (Best-of-N)



Inference-time optimization (Thompson sampling)

Beta(α, β) distribution

$$E[p] = \frac{\alpha}{\alpha + \beta}$$



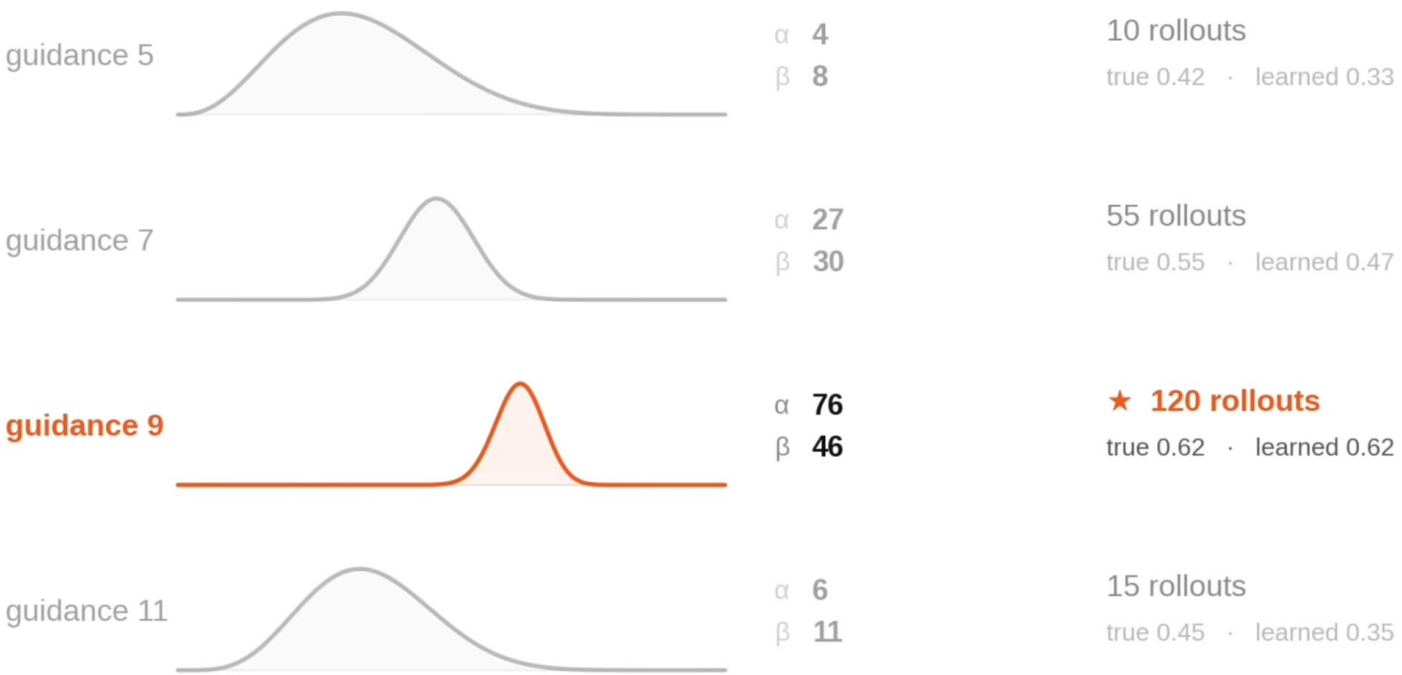
α (successes) 56

β (failures) 24

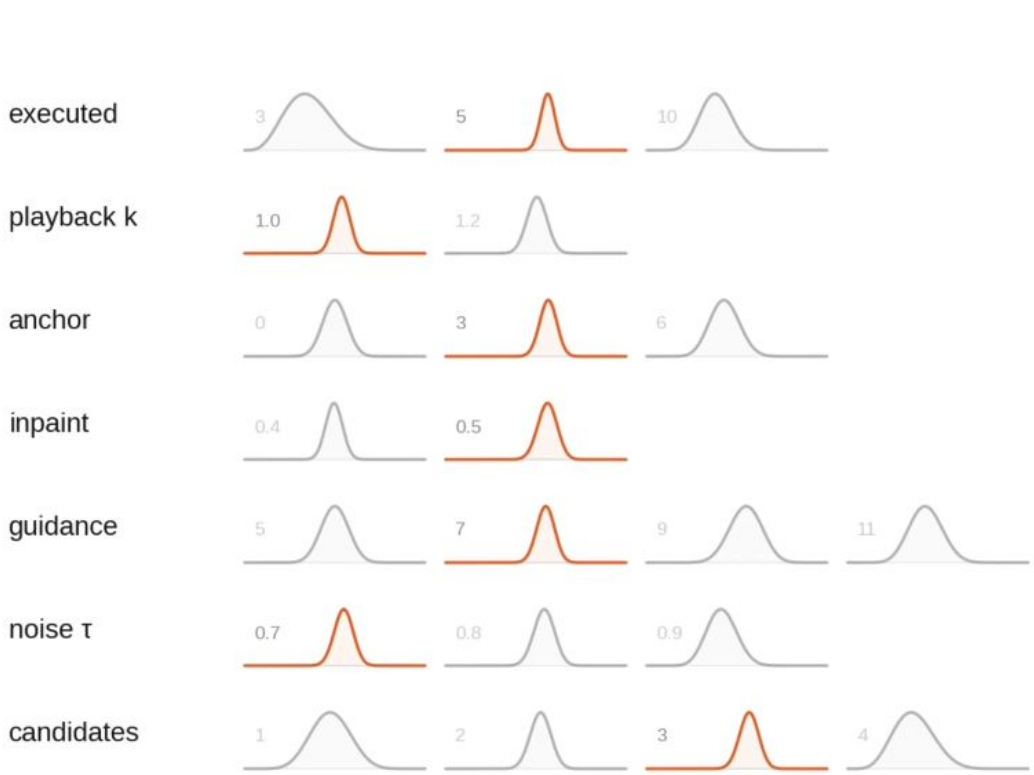
✓

rollouts: 78

Inference-time optimization (Thompson sampling)



Inference-time optimization (Thompson sampling)



	top long	top short	pant long	pant short
executed	5	5	3	3
playback k	1.0	1.0	1.0	1.0
anchor	6	3	3	3
inpaint	0.4	0.4	0.5	0.5
guidance	7	7	9	7
noise τ	0.9	0.7	0.7	0.7
candidates	2	3	3	3

Results













Discussion

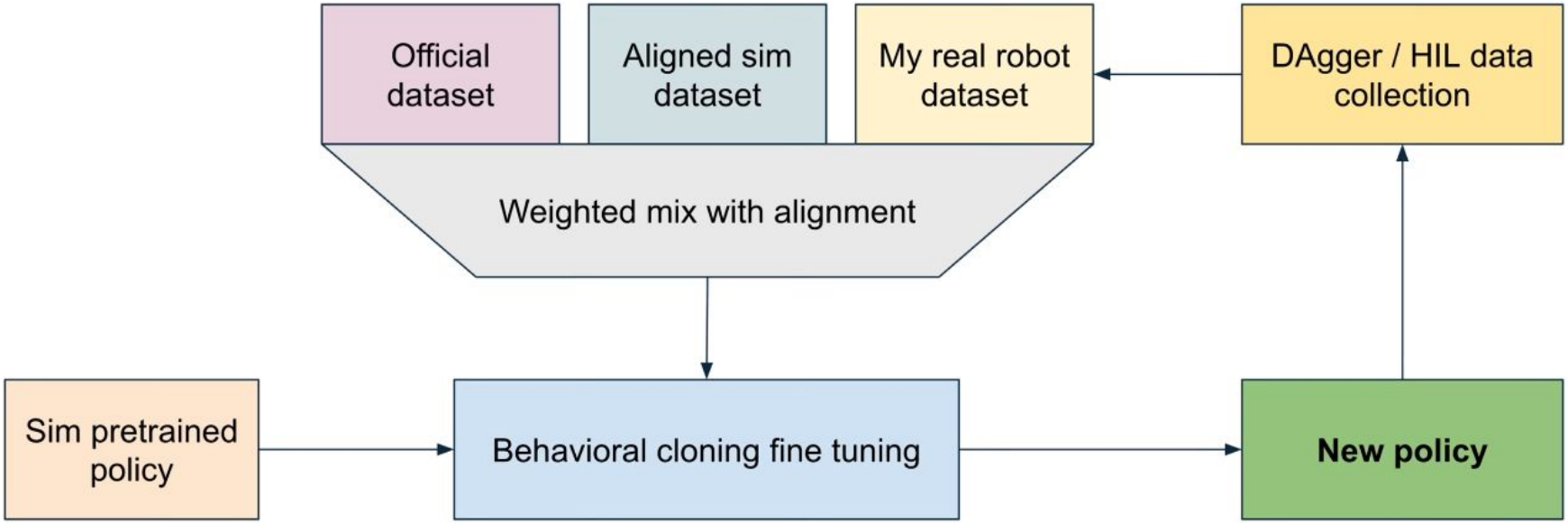
Worked well:

- AWR + RECAP
- Reward and advantage
- Async pipeline
- Inference time optimization

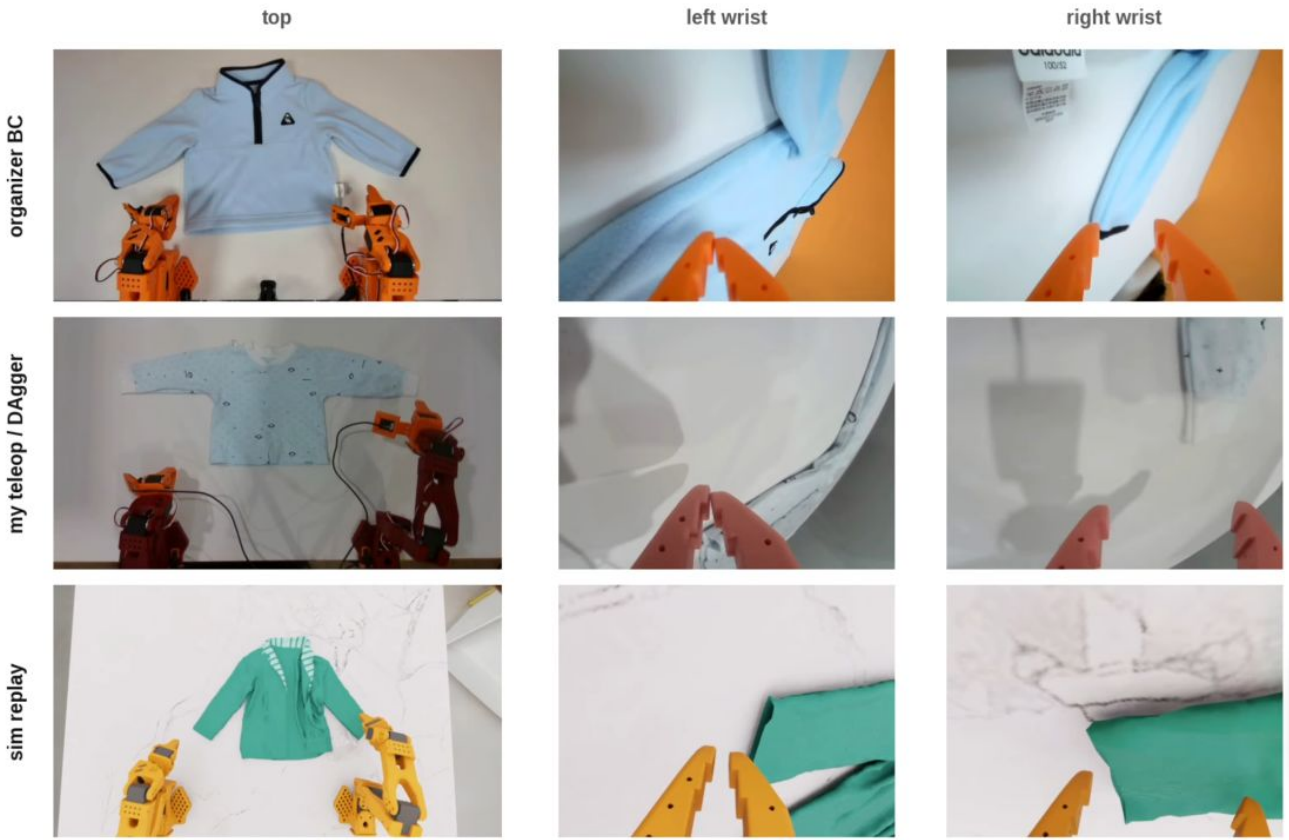
Challenges:

- HIL recovery
- Simulation artifacts
- Sample efficiency

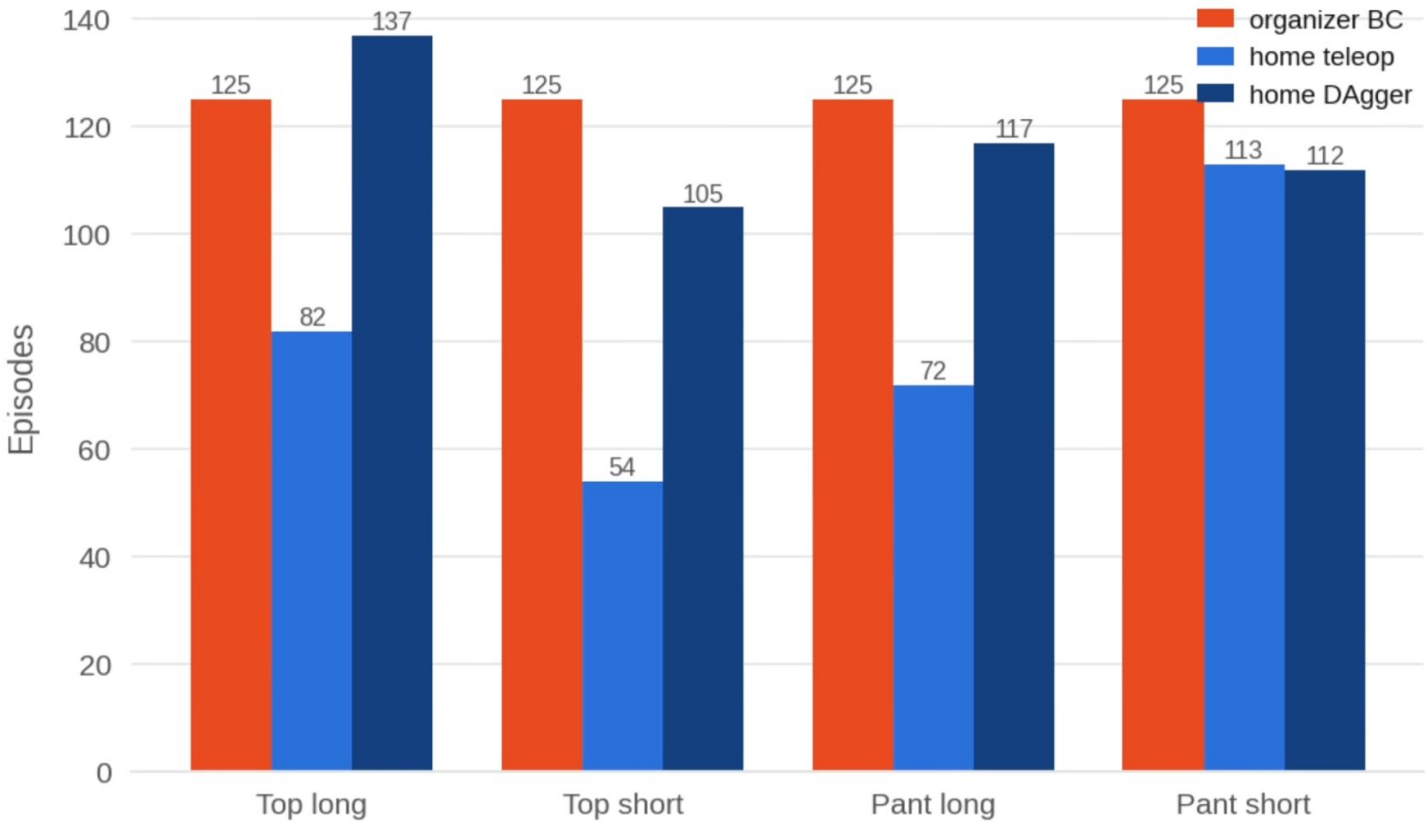
Sim 2 Real transfer: Data



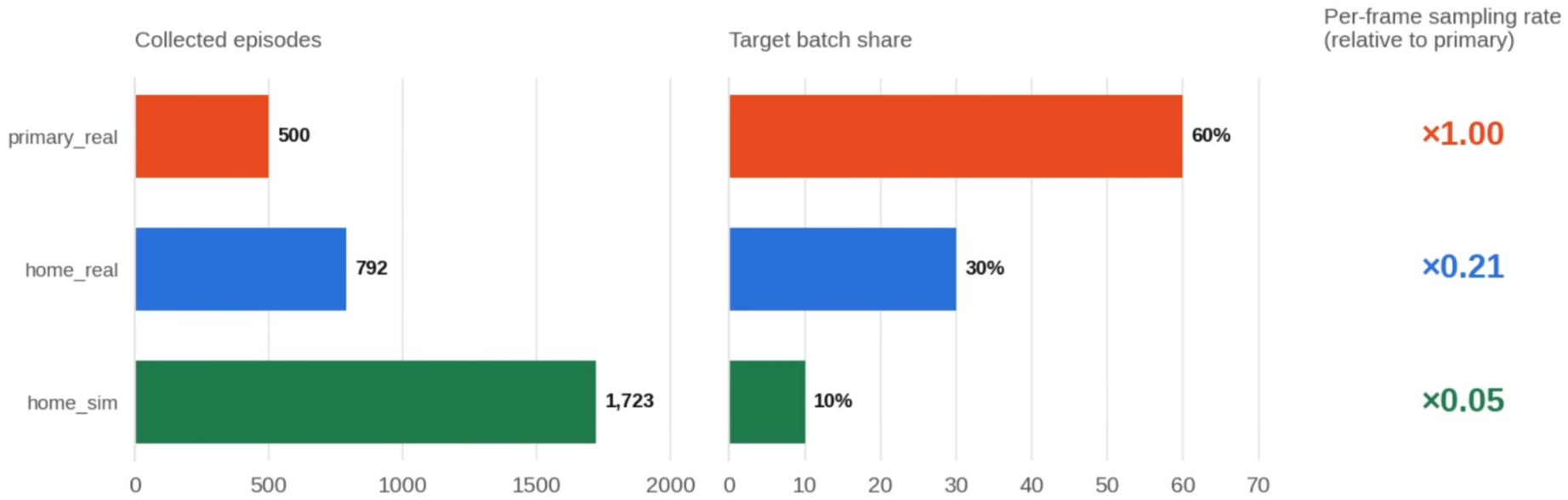
Sim 2 Real transfer: Data



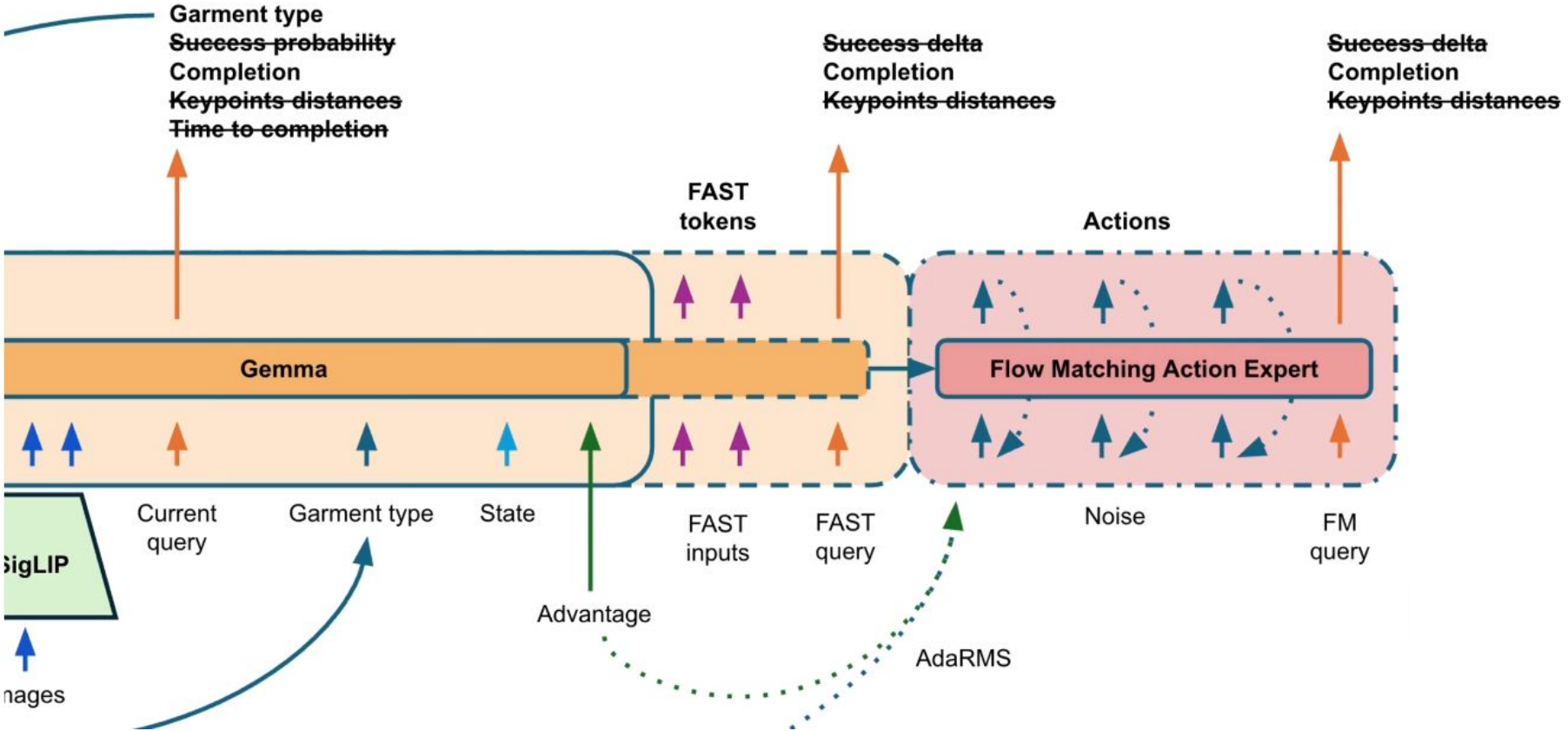
Sim 2 Real transfer: Data



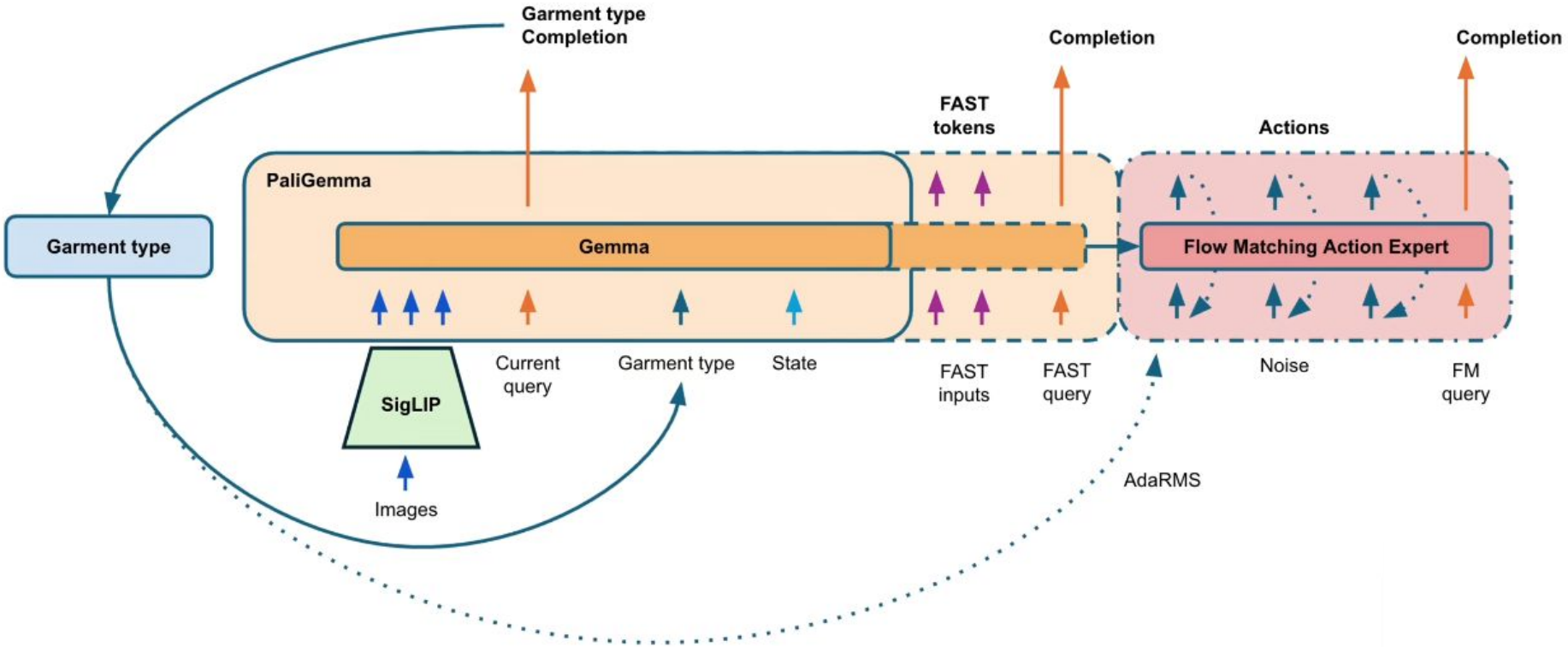
Sim 2 Real transfer: Data



Sim 2 Real transfer: Model



Sim 2 Real transfer: Model



Sim 2 Real transfer: Setup



https://www.youtube.com/watch?v=0j2B2cXoiu4

Sim 2 Real transfer: Setup

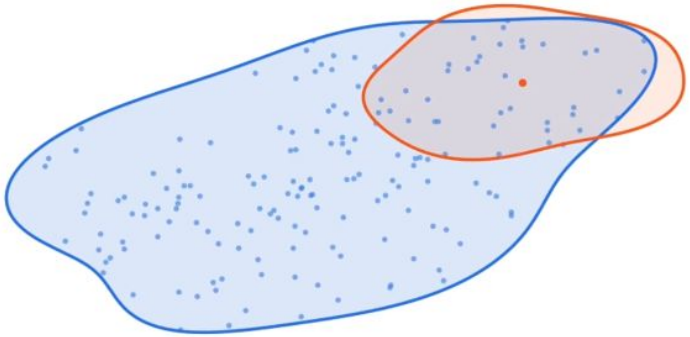


https://www.youtube.com/watch?v=0j2B2cXoiu4

Sim 2 Real transfer: Data alignment

1 align

2 diversify



training distribution

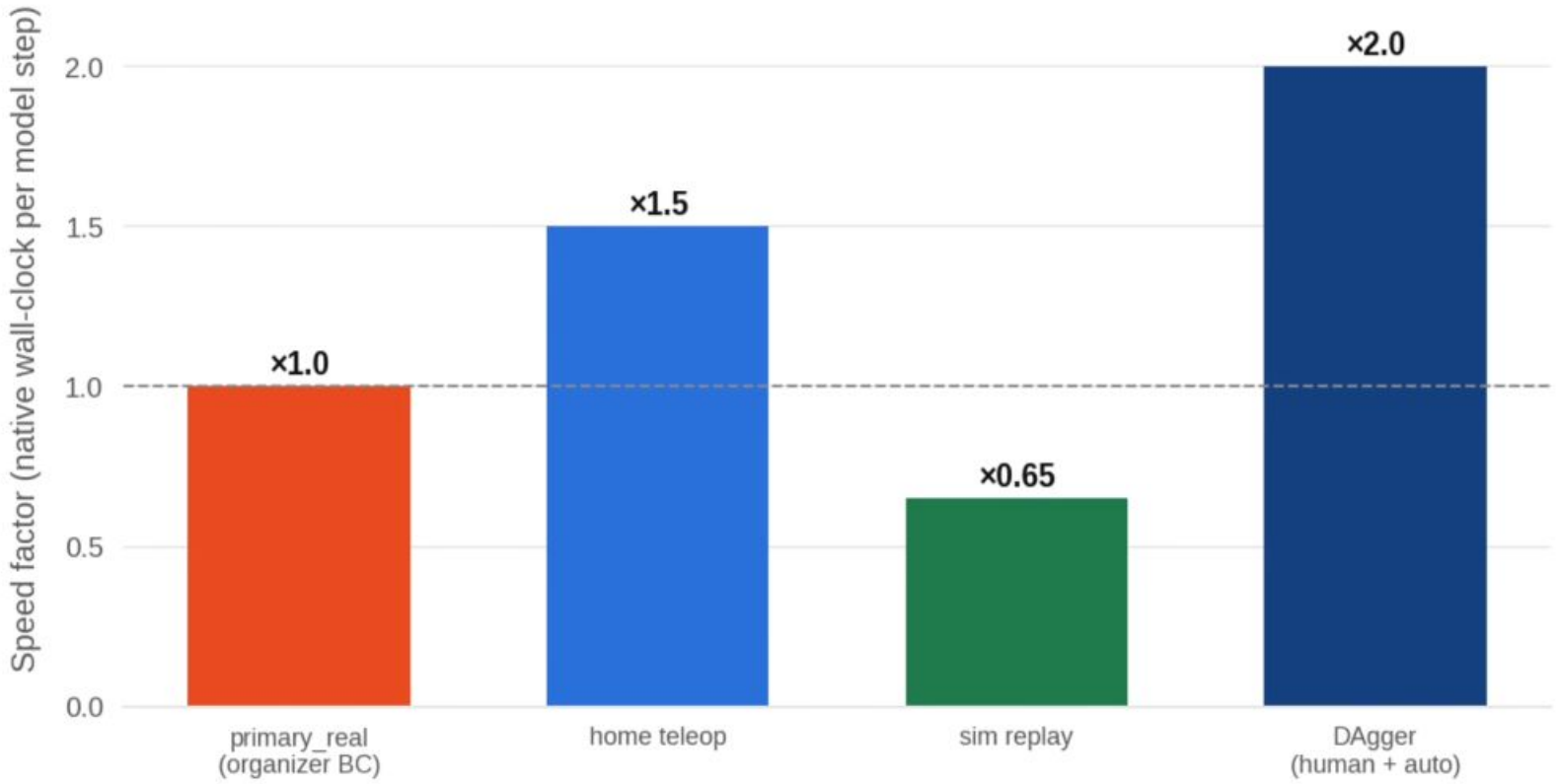
evaluation rig

Sim 2 Real transfer: Data alignment



https://www.youtube.com/watch?v=0j2B2cXoiu4

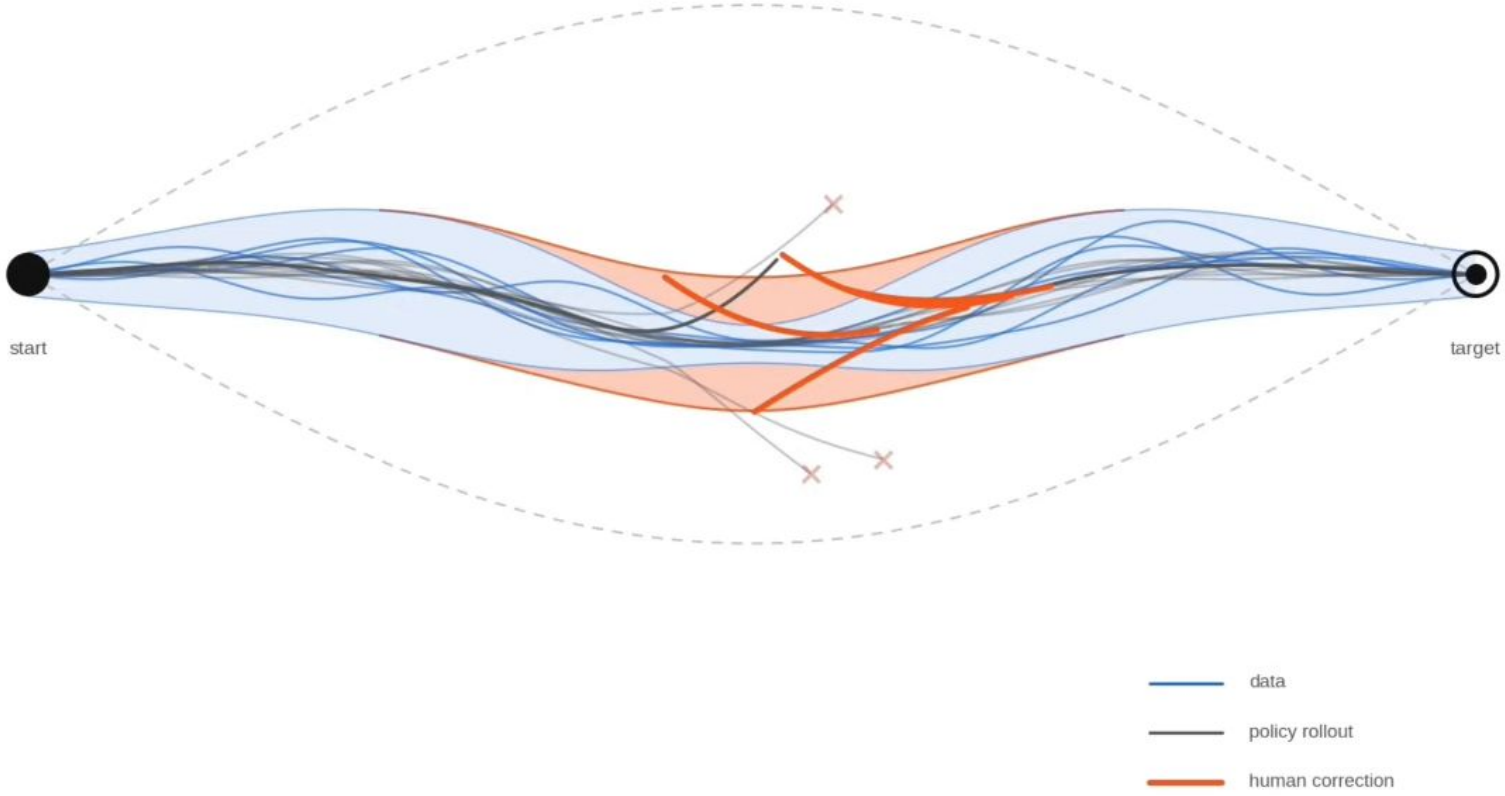
Sim 2 Real transfer: Data alignment



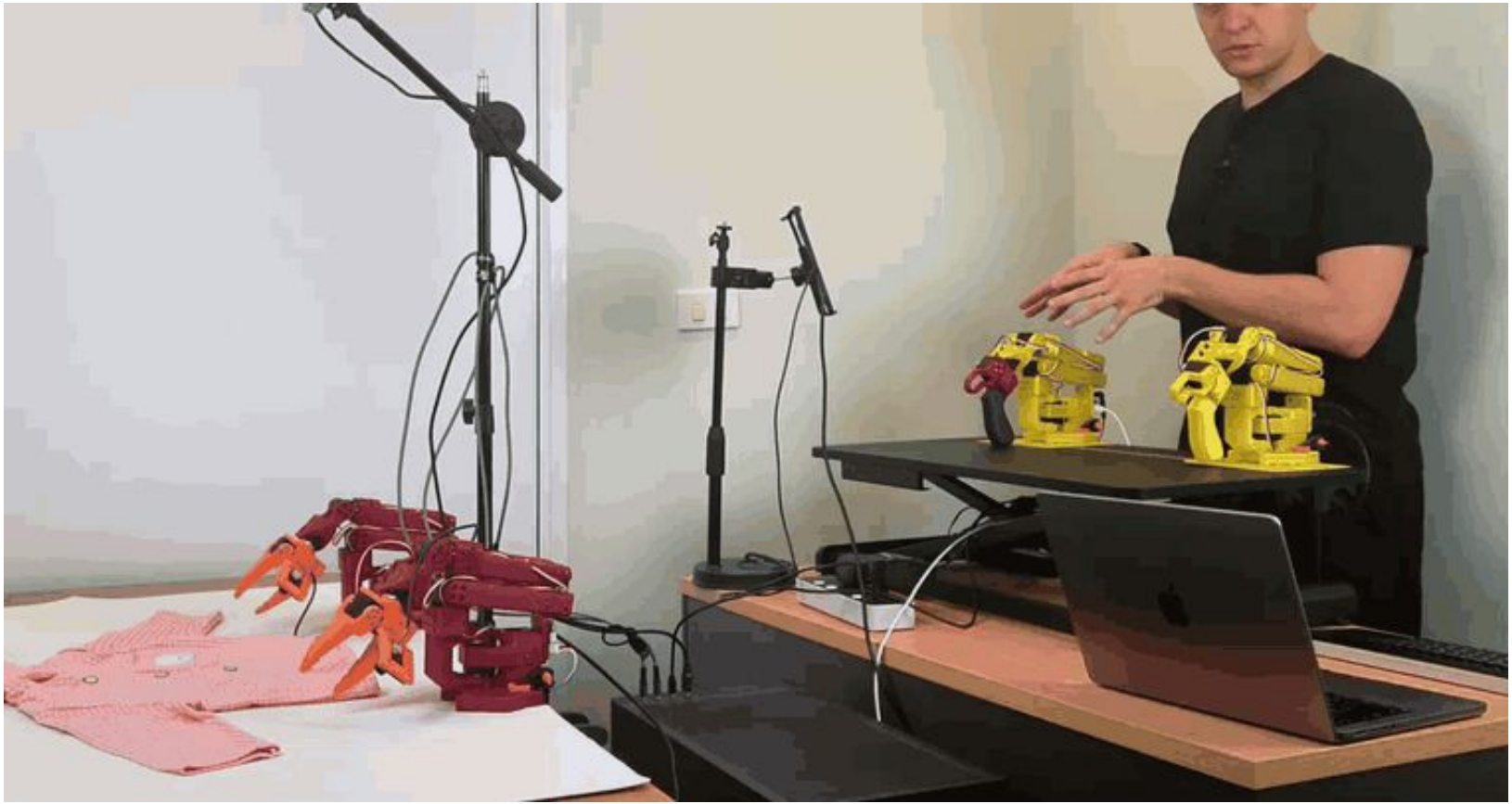
Sim 2 Real transfer: Data augmentation



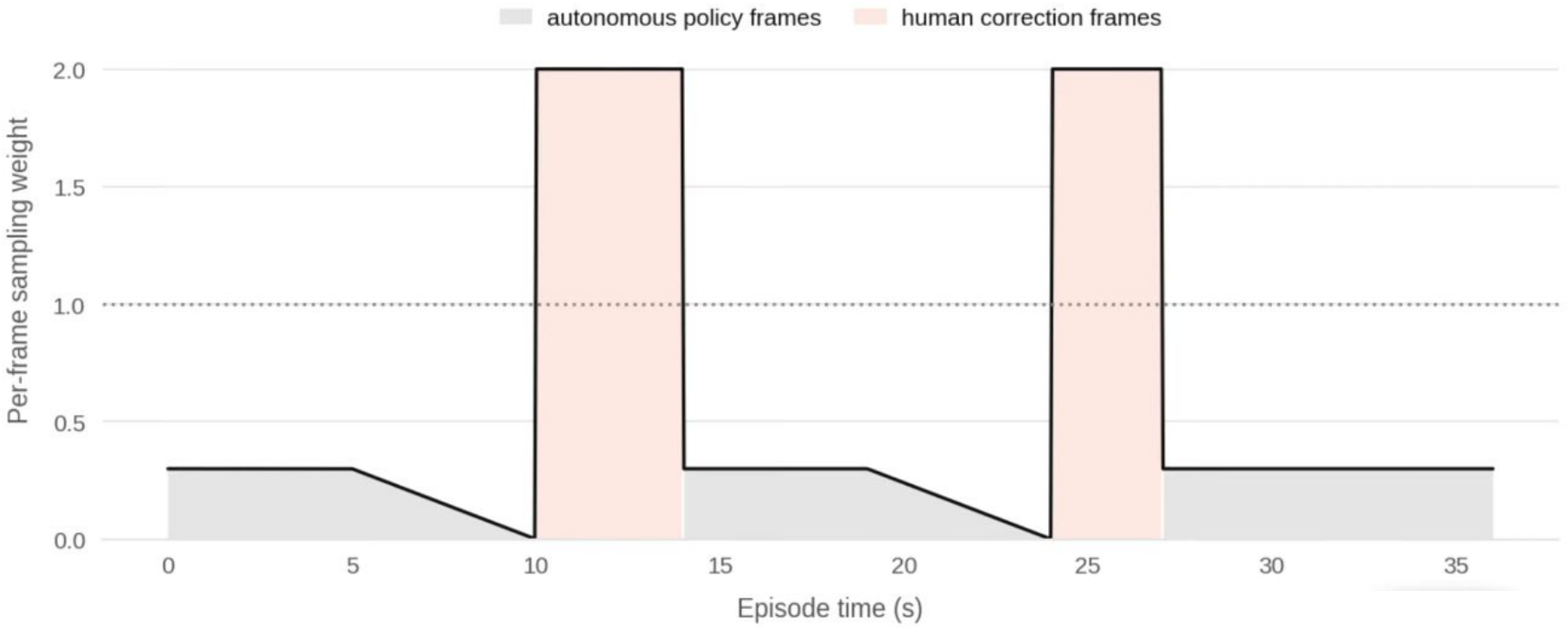
DAgger: Human-in-the-loop



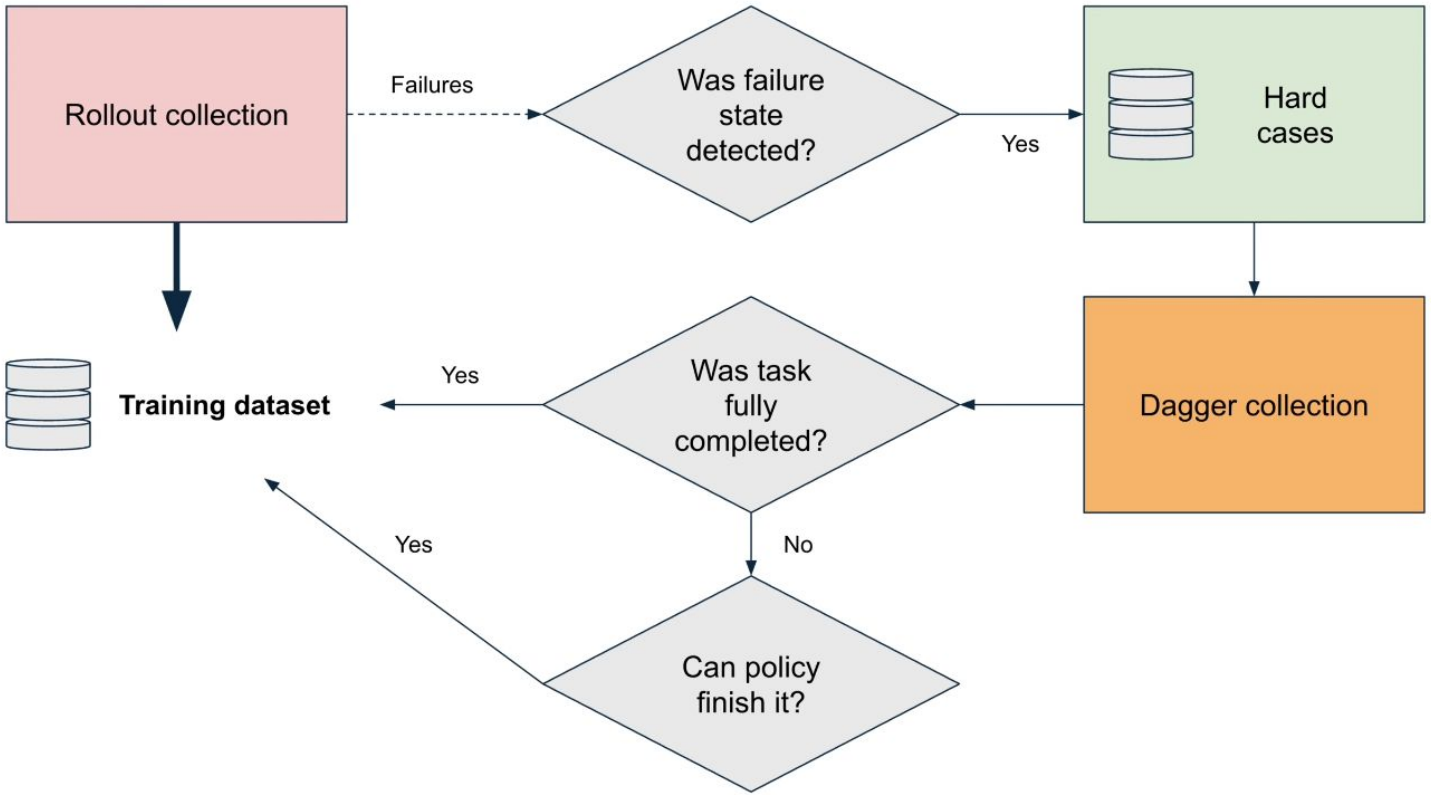
DAgger: Human-in-the-loop



DAgger: Human-in-the-loop



DAgger: Human-in-the-loop (in a sim)



Results (challenge)

