Learning to Adapt to Humans Without Human Data

Anonymous authors
Paper under double-blind review

Abstract

Any intelligent agent operating in an open-ended environment with multiple co-players must be able to adapt to behaviors it has not seen before. In safety-critical settings such as self-driving, this adaptation should happen continuously and rapidly, as a function of recent context. We ask whether such adaptation can be learned without any human data. To this end, we train a memory-based driving policy with reinforcement learning against a diverse population of fully synthetic co-players. The resulting policies adapt over repeated encounters with held-out human driving replays, improving their performance as they gain experience. We ablate the training procedure and demonstrate the importance of memory. Analysis of the attention map reveals an interpretable mechanism behind the agent's adaptation. Our results suggest a path toward agents that not only generalize across diverse partners, but also adapt beyond their training population to entirely new co-players through interaction.

1 Introduction

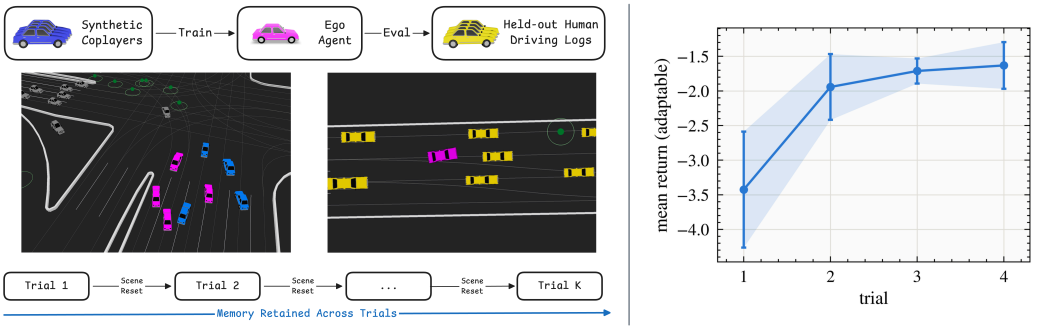


Figure 1: **Ego agent learns to adapt through repeated encounters. Left:** The ego policy is trained with synthetic co-players and evaluated on held-out human driving logs. Between trials, the scene and traffic trajectories reset while memory is retained. The two scenes illustrate each phase: during training (left), the egos (magenta) share the road with synthetic co-players (blue); during evaluation (right), all other vehicles replay held-out human logs (yellow). **Right:** Mean return across four trials for the entropy-bound 0.05 configuration. Mean across three training seeds on held-out adaptable maps; shading indicates ±1 standard deviation.

Autonomous agents operating in an open-ended environment will inevitably encounter other agents whose behavior they have never seen. Some are cautious, others aggressive. Some are predictable, others erratic. No training distribution can enumerate every partner an agent might meet, so the agent must adapt to new ones as it encounters them. Crucially, this adaptation ought to happen in real time. The agent should infer how its partners behave during the series of interactions it encounters and adjust while these interactions

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

are unfolding. In safety-critical settings, an agent that can only improve in one discrete step after a batch of many interactions pays a great cost for its repeated mistakes, as compared to one which adapts continuously in context. How to learn this adaptation capability is not obvious. Adaptation could be carried out through updates to the agent's weights at test time (Munkhdalai and Yu, 2017; Sun et al., 2020), or through its memory of recent experience (Duan et al., 2016; Team et al., 2023). In either case the agent must learn from its training partners a way of adapting that transfers to partners it has never met.

Standard multi-agent training does not produce adaptation, for two reasons. The first is the training distribution. Single-policy self-play in fully cooperative games produces homogeneous partners that converge to one specific coordination mechanism; scripted partners require manual design and cover only a narrow range of behaviors; and log replay provides realistic trajectories but these do not react to the agent. The second is the training objective. Even with diverse partners, agents are typically trained to perform well from the start of each interaction, and are never rewarded for improving as they learn more about the partner they face. Prior work in meta-reinforcement learning shows that adaptation emerges when both conditions are met, by training an agent with memory to improve across repeated attempts over a sufficiently diverse set of tasks (Duan et al., 2016; Team et al., 2023). In multi-agent settings, this requires a diverse population of partners together with a training procedure that rewards adapting to them.

To address these requirements, we introduce Adaptive Population Play (APP), a method for training agents that adapt in context to diverse and unknown co-player behaviors. We apply our method to autonomous driving, a safety-critical domain where other drivers' behaviors vary widely and cannot be observed in advance. Our approach consists of two stages. In the first stage, we train a population of co-players conditioned on reward, entropy and discount factors, which generates behavioral diversity without manual specification. High entropy co-players exhibit exploratory and less predictable driving, while low discount co-players exhibit myopic behavior that prioritizes immediate rewards. By varying these factors, we produce a spectrum of driving styles, with direct control over the diversity of the population.

In the second stage, we train an adaptive ego agent against this diverse population. Each episode consists of repeated trials of the same scene, and the ego retains its memory across trials. The adaptive agent uses this temporal context to observe co-player behavior over multiple trials, implicitly infer their driving style, and adjust its policy accordingly. Crucially, the ego agent receives no explicit information about the co-player's conditioning. It must adapt from its interactions alone, matching real-world deployment constraints where other drivers' intentions are not observable. We train the ego with PPO (Schulman et al., 2017) using a transformer architecture for the policy network. Each episode of repeated trials is a meta-learning problem, in which the policy learns how to use its own history to act better in later trials.

We evaluate the ego on held-out scenes in which every other vehicle replays a recorded human trajectory, a type of partner never seen during training. The ego's performance improves over successive trials without any parameter updates (Figure 1).
In summary, we make the following contributions:

1. **Adaptive Population Play (APP)**. We introduce a two-stage method that trains in-context adaptation to unseen co-players using only a synthetic population.
2. **Transfer to human drivers.** Adaptation learned with synthetic co-players carries over to held-out human driving, improving performance over repeated encounters.
3. **Interpretable adaptation.** Cross-trial attention reveals which experiences drive changes in behavior, essential for trusting adaptive agents in safety-critical settings.

2 Adaptive Population Play

We train a driving policy to use experience from earlier encounters when acting in the same scene again. Our approach combines a conditioned co-player population, repeated trials with retained memory, and a transformer policy trained through reinforcement learning.

2.1 Driving Environment

We study multi-agent adaptation in PufferDrive (Cornelisse, Cheng, et al., 2025), a fast, data-driven driving simulator built on the PufferLib reinforcement learning framework (Suarez, 2024) that enables large-scale multi-agent training on a single GPU. To ground the simulation in realistic scenarios, we use the nuPlan dataset (Caesar et al., 2022), drawing on its recorded driving logs to initialize road layouts, starting vehicle states, and goal locations.

We distinguish two roles for vehicles:

- **Ego agent** is a vehicle controlled by the adaptive policy we train and evaluate.
- **Co-players** are the other policy-controlled vehicles that share the scene with the ego and generate its surrounding traffic.

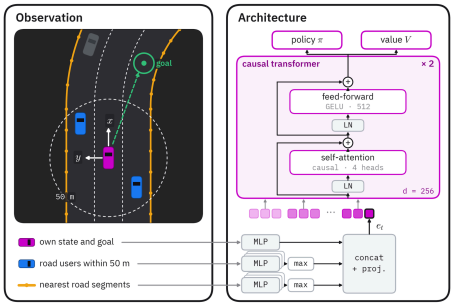


Figure 2: **Each policy encodes an egocentric observation and attends over the whole episode.** Co-players share this architecture and additionally receive the conditioning vector z.

For each ego, we define a goal-conditioned navigation task. The ego must reach its assigned goal while avoiding collisions and going off-road. During training, the ego interacts with co-players controlled by a frozen co-player policy, whose construction we describe in Section 2.2. During evaluation, the co-players are replaced by vehicles replaying recorded human trajectories from held-out scenes.

Each policy acts from an egocentric observation of its own state, relative goal position, surrounding traffic and road geometry. The observation contains nine ego features, up to 63 neighboring vehicles and 200 road segments.

The action space is a 91-way categorical distribution over 7 acceleration and 13 steering values. The policy acts in closed loop on these state observations. Figure 1 summarizes how we use this environment. We formalize the construction of the co-player population and the repeated-trial formulation in Section 2.2 and Section 2.3. Example renders are available for a training scene[1] and an evaluation scene[2].

2.2 Building a Co-player Population

We construct the co-player population by training a driving policy under various reward weights, discount factors, and entropy coefficients. The policy receives these parameters as a conditioning vector, $z = (w_{\text{goal}}, w_{\text{collision}}, w_{\text{offroad}}, \beta, \gamma)$, allowing one network to represent behaviors learned under different objectives. This differs from earlier work on in-context adaptation, where diversity typically comes from varying the task in a single-agent environment (Duan et al., 2016; Team et al., 2023; Team et al., 2021; Wang et al., 2017; Zintgraf et al., 2020), while work that varies partners has mostly targeted zero-shot coordination rather than adaptation (Cusumano-Towner et al., 2025; Strouse et al., 2022). In our setting, the task is fixed (goal-reaching) and the diversity lies in the other agents, which a deployed driving policy must adapt to across multiple interactions.

During co-player training, the three reward weights determine the reward assigned to each transition:

$$r_t(z) = w_{\text{goal}} g_t + w_{\text{collision}} c_t + w_{\text{offroad}} b_t + r_t^{\text{lane}}. \tag{1}$$

Here, g_t, c_t, and b_t are binary indicators for reaching the goal, colliding with another vehicle, and going off-road at step t, and r_t^{lane} is the lane-alignment shaping term. Increasing the

[1] https://youtube.com/shorts/2uk_VlztFWA
[2] https://youtube.com/shorts/p6owzdmcgSQ

goal weight increases the reward for task completion, while more negative collision and off-road weights increase the corresponding penalties. These rewards are used only to train the co-player and are separate from the ego's reward.

We use these rewards and the sampled discount γ to compute advantage estimates $\hat{A}^\gamma$ and value targets $\hat{R}^\gamma$. The co-player is optimized using PPO, minimizing

$$L(\varphi) = L_{\text{PPO}}\big(\varphi; \hat{A}^\gamma\big) + c_V L_V\big(\varphi; \hat{R}^\gamma\big) - \mathbb{E}_t\Big[\beta_t \mathcal{H}\big(\pi_\varphi(\cdot \mid x_t, z_t)\big)\Big]. \tag{2}$$

The first term is the clipped policy loss, which encourages actions with positive estimated advantage while limiting changes from the rollout policy. The second is the value-function loss, which trains the critic to predict the return targets, where c_V is the value-loss coefficient. The final term is an entropy bonus weighted by the sampled coefficient β_t, where x_t denotes the observation history through the current step. Together, the reward weights determine which outcomes are valuable, γ controls how much future rewards are weighted, and β controls how strongly the policy is encouraged to stay stochastic during training.

After training, we freeze the co-player policy. During ego training, we sample each conditioning parameter independently and uniformly from the ranges in Table 1. Co-players within a scene share the sampled vector, which remains fixed across repeated trials. The co-players receive this vector as part of their observation, but the ego does not.

Table 1: **Co-player conditioning ranges.** Sampled independently and uniformly per scene; $e \in \{0.05, 0.10, 0.20, 0.50\}$.

Parameter	Sampling range
Goal reward weight	$[0, 1]$
Collision reward weight	$[-2, 0]$
Off-road reward weight	$[-2, 0]$
Entropy coefficient β	$[0, e]$
Discount factor γ	$[0.4, 1]$

2.3 Repeated-Trial Formulation

We organize experience into episodes containing k trials of the same driving scene, following the multi-trial structure used for in-context adaptation in prior work (Duan et al., 2016; Team et al., 2023). A trial is one attempt to reach the assigned goal. Across trials, the map, initial vehicle states, goal locations, and co-player conditioning remain fixed. At each trial boundary, the simulator restores every vehicle to its initial state, but the ego retains its observation history, so the embeddings of all earlier trials remain in its context (Figure 2, right).

Let o_t denote the ego's observation at timestep t, with time indexed across the episode. Its action is sampled according to

$$a_t \sim \pi_\theta(\cdot \mid o_t, h_t), \qquad h_t = (o_0, ..., o_{t-1}). \tag{3}$$

At evaluation, the policy parameters θ remain fixed, so any improvement across trials must come from how the policy uses this history. A trial ends when all ego vehicles in the scene have reached their goals or when the 201-step time limit is reached. An ego that reaches its goal early is removed from the scene until the next trial begins.

2.4 Memory Architecture

The ego uses a decoder-only transformer (Vaswani et al., 2017) to retain information across trials (Figure 2). Separate encoders process its own state and goal, the surrounding road users, and the road segments, with road users and segments encoded per object and max-pooled. The three outputs are combined into one embedding e_t per timestep. The transformer has two pre-LN layers (Xiong et al., 2020), each applying causal multi-head self-attention followed by a feed-forward network, and attends over the embeddings of the current and all preceding trials. Policy and value heads map its output at the current step to action logits and a value estimate.

The context spans the full episode of up to $201 \times k$ steps, and positional indices continue across trial boundaries rather than restarting. Prior work typically gives the agent its

previous action and reward, and often an explicit trial or episode boundary signal (Duan et al., 2016; Team et al., 2023; Wang et al., 2017). Our policy receives none of these. It observes only the scene, and must infer from the reset that a new trial has begun and from its own history how the previous attempt went.

2.5 Training the Adaptive Ego

We train the ego with PPO against the frozen co-player population. Half of the policy-controlled vehicles in each episode are egos and half are co-players. Egos share parameters but maintain separate observation histories. For each scene, we sample a conditioning vector for the co-players and run up to k trials with the same scene and conditioning. The ego's reward is fixed throughout training, with $+1$ for reaching the goal, -0.5 for collisions and off-road events, and a lane-alignment shaping term with coefficient 0.05.

Although attention spans all trials, we compute value targets within each trial, stopping bootstrapping and the advantage recursion at trial boundaries. Later-trial losses therefore train the policy to use its earlier experience, but rewards from later trials are never credited to actions taken in earlier ones. This differs from prior work, which either treats the multi-trial episode as a single continuous return (Duan et al., 2016; Team et al., 2023) or explicitly propagates credit across attempts with a cross-episode discount (Jiang et al., 2026). Those choices reward exploration in early trials that pays off later. Our objective does not, so any adaptation that emerges comes from exploiting experience rather than deliberately gathering it, a choice we make with safety in mind.

3 Experiments and Results

3.1 Evaluation Setup

We evaluate each trained ego on 539 held-out nuPlan scenes that were excluded from its training pool. In every scene, a single vehicle is controlled by the ego policy, and all other vehicles replay their recorded human trajectories, a type of partner the ego never encountered during training. Each episode consists of $k = 4$ trials unless stated otherwise, and the policy weights remain fixed throughout. We run 20 rollouts per scene for each of three independently trained policies, and report means over rollouts, scenes and training seeds.

We measure adaptation as the change in return from the first trial to the last, $\Delta R = R_k - R_1$, where R_i is the mean return in trial i. Most held-out scenes leave little room for improvement, because the ego already reaches its goal reliably in the first trial. We therefore report ΔR on *adaptable* scenes, defined for each policy as those where its first-trial success rate over the 20 rollouts is below 0.8. Because this selection favours scenes with a poor first trial, part of the measured improvement can arise from chance alone. The memory-reset control in Section 3.3 estimates this effect directly.

3.2 Adaptation to Held-out Human Scenes

We first ask whether an ego trained only against synthetic co-players adapts when it faces human drivers. Figure 3 shows the ego's mean return on adaptable scenes across the four trials of an episode, for four co-player populations that differ in their entropy bound e. In every population, return rises over successive trials, and the gain is largest when co-players are trained with a low entropy bound. We hypothesize that setting the entropy bound of the co-players too high leads to behavior that is too erratic for the ego agent to be able to infer in-context. For reference, a policy trained with single-trial episodes, which never learns to use earlier attempts, performs at a similar level to the first trial on these scenes (violet dashed line).

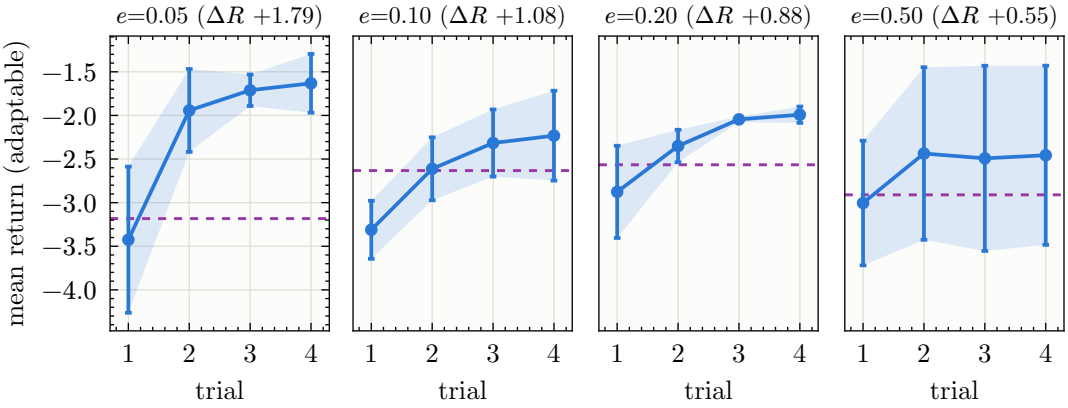


Figure 3: **The ego improves over repeated trials.** Mean return on adaptable scenes per trial for each co-player entropy bound e; titles give ΔR. Mean over three seeds, shading ± 1 SD. The violet dashed line is a policy trained with single-trial episodes ($k = 1$).

3.3 Adaptation Requires Memory

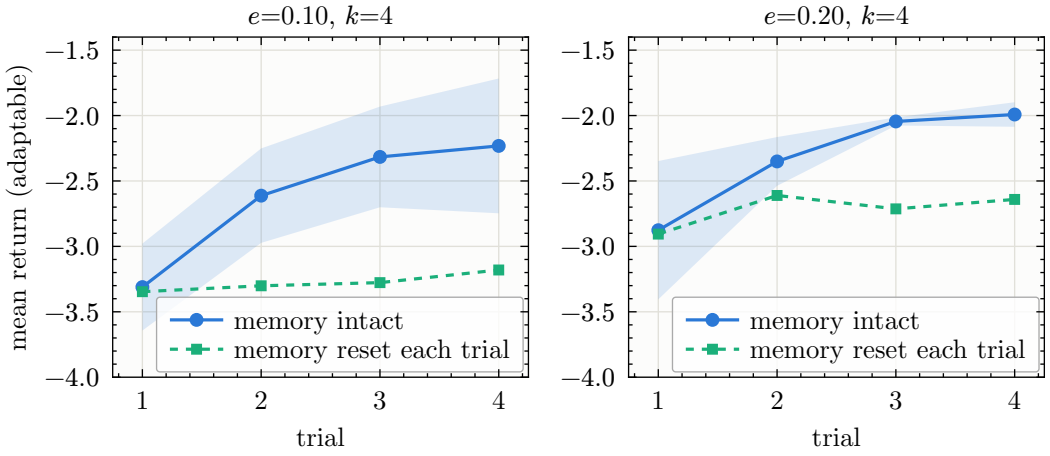


Figure 4: **Clearing memory between trials removes most of the improvement.** Mean return on adaptable scenes per trial with memory intact (solid) and reset at each trial boundary (dashed), for $e = 0.10$ and $e = 0.20$. Mean over three seeds.

Since the ego's policy parameters remain fixed throughout, adaptation can only come from how the ego uses its memory of earlier trials. To test how the improvement depends on memory, we evaluate the same trained policies with their history cleared at every trial boundary. Each trial then begins with an empty context, exactly like the first, while the scene, co-players and policy parameters are unchanged. We evaluate this control on the same adaptable scenes as the intact policy, so any improvement it shows reflects chance and scene selection rather than experience. Figure 4 compares the two. With memory intact, return climbs across trials, while with memory reset it stays close to its first-trial level. The adaptation observed in Section 3.2 therefore depends on the ego's memory of earlier trials, not on repetition of the scene alone.

3.4 Adaptation Decomposition

To understand what changes between trials, we split ΔR into the reward components that make it up (Figure 5). Across both co-player populations, most of the improvement comes from fewer collisions with other vehicles, while off-road events contribute comparatively little. The ego also reaches its goal more often in later trials, although this accounts for

a smaller share of the gain, and lane alignment changes little. The ego therefore adapts primarily by learning, from its earlier attempts, how to avoid conflicts with the surrounding traffic.

Because this adaptation emerges from the ego's fixed reward, its composition likely reflects the relative weighting of the reward terms. We expect that changing this weighting, for example by increasing the goal reward or adding shaping for progress, would shift what the ego adapts to, making the reward a lever for controlling the kind of adaptation that emerges.

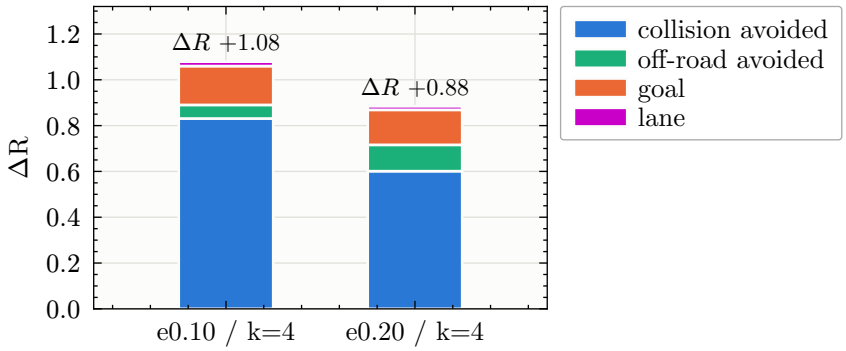


Figure 5: **Most of the improvement comes from avoiding collisions with other vehicles.** Change in each reward component from the first to the last trial on adaptable scenes. Mean over three seeds.

3.5 Interpreting Adaptation Through Attention

In safety-critical settings, it is not enough for a vehicle to change its behavior. We also need to understand why it did so, and which experience caused the change. Because the ego's memory is implemented through attention, we can inspect which earlier moments the policy draws on at each step, and relate its adaptation to specific events in its history. We illustrate this on a single evaluation episode with three trials (Figure 6). In the first trial, the ego approaches its goal and collides with oncoming traffic. In the second, it approaches earlier, abandons the approach, and collides again. In the third, it approaches slowly, waits behind a vehicle in the adjacent lane, and reaches its goal. The attention map traces this change in behavior back to the earlier failures. Most attention falls near the diagonal, on the most recent timesteps, as the ego tracks the current state of the scene. Attention to earlier trials, in contrast, is highly selective. Throughout the second trial, the ego repeatedly attends to its first-trial collision, visible as a vertical band in the attention map. In the third trial, it attends to both the first-trial collision and the failed approach of the second, so the retrieved information accumulates across attempts.

The ego also attends to the frames leading up to each earlier collision, retrieving not only the failure itself but the decisions that produced it. This retrieval begins at the start of each trial, before the ego reaches the location of its earlier failure, while the rest of each earlier trial receives little attention. Retrieval also stops once it is no longer useful. In the second trial, attention to the first-trial collision ends once the ego has passed its goal and can no longer reach it.

These patterns are consistent with the ego's cautious approach in the third trial being driven by the collisions it experienced earlier, which it retrieves while deciding how to act. More broadly, attention over past trials offers a way to audit an adaptive policy, identifying which experiences caused a change in behavior and verifying that the policy adapts for the right reasons.

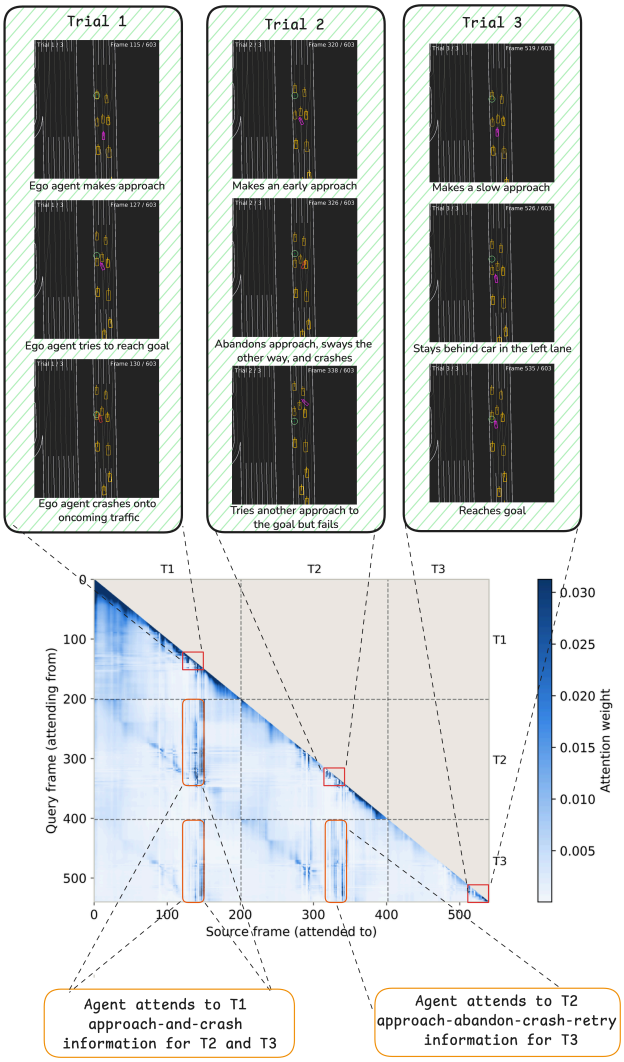


Figure 6: **Attention reveals which experiences drive the ego's change in behavior.** Top: three trials of one held-out scene. Bottom: attention from each timestep (rows) to earlier timesteps (columns); dashed lines mark trial boundaries. Boxes highlight selected events and cross-trial attention regions. Model trained with $k = 3$. Full render - https://youtu.be/nMsiyyTZ_fM

4 Related Work

4.1 Generalizing to Unseen Partners

Acting alongside unseen partners, including humans, has been studied as zero-shot coordination and ad hoc teamwork (Hu et al., 2021; Mirsky et al., 2022; Wang et al., 2026). Behavioral models of humans improve coordination with people but require human data (Carroll et al., 2020), so most methods instead train a best response to a diverse population of synthetic partners (Lupu et al., 2021; Strouse et al., 2022; Yu et al., 2023; Zhao et al., 2022). Diversity is generated from training checkpoints (Strouse et al., 2022), explicit diversity objectives (Lupu et al., 2021; Zhao et al., 2022), or randomized rewards and social preferences (McKee et al., 2020; Tang et al., 2021; Yu et al., 2023), usually as separate policies but occasionally within a single conditioned one (Baker, 2020). In all cases, the

agent is optimized to perform well from the start of an interaction rather than to improve through it, and adaptation at test time remains an open direction (Yu et al., 2023).

4.2 Learning to Adapt In Context

Meta-reinforcement learning offers such adaptation. Memory-based policies trained across many tasks learn to improve over repeated episodes without gradient updates (Beck et al., 2025; Duan et al., 2016; Wang et al., 2017), and transformer memory has extended this to long histories and open-ended task spaces (Elawady et al., 2024; Grigsby et al., 2024; Melo, 2022; Team et al., 2023). These agents typically adapt to changes in the task rather than in other agents, receive their previous reward and explicit trial boundaries as input (Duan et al., 2016; Team et al., 2023; Wang et al., 2017), and remain largely confined to small-scale problems. The few methods that adapt to partners in context have been demonstrated only with synthetic partners in simulated games (Charakorn et al., 2021; Team et al., 2023; Wang et al., 2026; Zintgraf et al., 2022), and in-context learners still struggle to adapt robustly in multi-agent settings (Jing et al., 2026). Analyses of their memory show that they attend across episodes (Elawady et al., 2024; Grigsby et al., 2024) but rarely connect this attention to the experiences that change their behavior.

4.3 Diverse Agents in Driving

Data-driven simulators now support multi-agent reinforcement learning on real traffic scenes at scale (Caesar et al., 2022; Cornelisse, Cheng, et al., 2025; Gulino et al., 2023; Kazemkhani et al., 2025; Vinitsky et al., 2023), and self-play in them produces reliable driving policies (Cornelisse, Pandya, et al., 2025; Cusumano-Towner et al., 2025), particularly when regularized toward human demonstrations (Cornelisse et al., 2026; Cornelisse and Vinitsky, 2024). Traffic diversity has been introduced through return conditioning (Rowe et al., 2024), social value orientation (Zhang et al., 2023), and a single policy conditioned on randomized reward weights (Cusumano-Towner et al., 2025), but the ego is trained to be robust to this diversity rather than to respond to it dynamically across multiple interactions. Work that adapts to other drivers has been limited to individual merging and intersection scenarios (Bouton et al., 2019; Hu et al., 2020; Ma et al., 2021).

5 Discussion and Limitations

We introduced Adaptive Population Play, a method that trains a driving policy to adapt in context to other agents without human driving data. Trained only against a conditioned co-player population, the ego agent improves over repeated trials in held-out scenes with human log replay. The attention map shows that the agent attends to its earlier failures when acting again.

Several limitations point to directions for future work. Because each trial replays the same scene with non-reactive human trajectories, we cannot separate adaptation to other drivers from memorization of the scene, and the human partners cannot respond to the ego as real drivers would. Evaluating against reactive, human-like partners (Cornelisse et al., 2026; Cornelisse and Vinitsky, 2024) and varying the co-player between trials would test adaptation to dynamic drivers. Resetting the world at trial boundaries is clearly unrealistic. Extending our approach to realistic trials with repeated structure but no hard resets would align the method better with deployment. One could train towards this with a curriculum of partial resets, for instance. Furthermore, since credit is cut at trial boundaries, the ego is never rewarded for gathering information early, so its adaptation reflects recall rather than exploration. Propagating credit across trials (Jiang et al., 2026) could turn this recall into active information gathering, but work would be needed to ensure that information gathering was done safely.

Our attention analysis is qualitative and does not establish that the retrieved memories cause the change in behavior. Intervening on the memories the ego attends to would test this directly, making attention a tool for auditing adaptive vehicles.

6 Statements

6.1 AI Disclosure

We used generative AI tools to write a large part of the code, to provide feedback on experimental methodology, to support the analysis and interpretation of results including the attention maps, and to create figures. We also used them to search for and summarize related literature, format references, and draft and edit parts of the paper. The research idea, hypotheses and core method were developed by the authors, and no AI tools were used to generate datasets or for translation. All AI-assisted code, text and literature descriptions were reviewed by the authors, every cited paper was checked against its source, and every reported number was checked against our own evaluation outputs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

6.2 Ethics Statement

This work involves no human subjects. All driving scenes come from the publicly available nuPlan dataset (Caesar et al., 2022), used under its terms of use. Our policies are trained and evaluated only in simulation, against non-reactive replays of human driving, and are not validated for real-world deployment. Any use of in-context adaptation in real vehicles would require extensive safety validation beyond this work.

6.3 Reproducibility Statement

The full method, training setup and evaluation protocol are described in the paper. Code for training and evaluation will be released upon publication.

References

Baker, B. *Emergent Reciprocity and Team Formation from Randomized Uncertain Social Preferences*, 2020. https://arxiv.org/abs/2011.05373

Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., and Whiteson, S. *A Tutorial on Meta-Reinforcement Learning*, 2025. https://doi.org/https://doi.org/10.1561/2200000080

Bouton, M., Nakhaei, A., Fujimura, K., and Kochenderfer, M. J. *Cooperation-Aware Reinforcement Learning for Merging in Dense Traffic*, 2019. https://arxiv.org/abs/1906.11021

Caesar, H., Kabzan, J., Tan, K. S., Fong, W. K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., and Omari, S. *NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles*, 2022. https://arxiv.org/abs/2106.11810

Carroll, M., Shah, R., Ho, M. K., Griffiths, T. L., Seshia, S. A., Abbeel, P., and Dragan, A. *On the Utility of Learning about Humans for Human-AI Coordination*, 2020. https://arxiv.org/abs/1910.05789

Charakorn, R., Manoonpong, P., and Dilokthanakul, N. *Learning to Cooperate with Unseen Agent via Meta-Reinforcement Learning*, 2021. https://doi.org/https://doi.org/10.5555/3463952.3464131

Cornelisse, D., and Vinitsky, E. *Human-compatible driving partners through data-regularized self-play reinforcement learning*, 2024. https://arxiv.org/abs/2403.19648

Cornelisse, Cheng, et al. *PufferDrive: A Fast and Friendly Driving Simulator for Training and Evaluating RL Agents*, 2025. https://github.com/Emerge-Lab/PufferDrive

Cornelisse, D., Hunt, J., Zhang, Z., Doulazmi, W., Joseph, K., Fisac, J. F., and Vinitsky, E. *Human-like autonomy emerges from self-play and a pinch of human data*, 2026. https://arxiv.org/abs/2606.19370

Cornelisse, Pandya, et al. *Building reliable sim driving agents by scaling self-play*, 2025. https://arxiv.org/abs/2502.14706

Cusumano-Towner, M., Hafner, D., Hertzberg, A., Huval, B., Petrenko, A., Vinitsky, E., Wijmans, E., Killian, T., Bowers, S., Sener, O., Krähenbühl, P., and Koltun, V. *Robust Autonomy Emerges from Self-Play*, 2025. https://arxiv.org/abs/2502.03349

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl ^2: Fast reinforcement learning via slow reinforcement learning. *Arxiv Preprint Arxiv:1611.02779*, 2016.

Elawady, A., Chhablani, G., Ramrakhya, R., Yadav, K., Batra, D., Kira, Z., and Szot, A. *ReLIC: A Recipe for 64k Steps of In-Context Reinforcement Learning for Embodied AI*, 2024. https://arxiv.org/abs/2410.02751

Grigsby, J., Fan, L., and Zhu, Y. *AMAGO: Scalable In-Context Reinforcement Learning for Adaptive Agents*, 2024. https://arxiv.org/abs/2310.09971

Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., Harb, J., Pan, X., Wang, Y., Chen, X., Co-Reyes, J. D., Agarwal, R., Roelofs, R., Lu, Y., Montali, N., Mougin, P., Yang, Z., White, B., Faust, A., … Sapp, B. *Waymax: An Accelerated, Data-Driven Simulator for Large-Scale Autonomous Driving Research*, 2023. https://arxiv.org/abs/2310.08710

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. *"Other-Play" for Zero-Shot Coordination*, 2021. https://arxiv.org/abs/2003.02979

Hu, Y., Nakhaei, A., Tomizuka, M., and Fujimura, K. *Interaction-aware Decision Making with Adaptive Strategies under Merging Scenarios*, 2020. https://arxiv.org/abs/1904.06025

Jiang, M., Grefenstette, E., and Rocktäschel, T. *Prioritized Level Replay*, 2021. https://arxiv.org/abs/2010.03934

Jiang, Y., Jiang, L., Teney, D., Moor, M., and Brbic, M. *Meta-RL Induces Exploration in Language Agents*, 2026. https://arxiv.org/abs/2512.16848

Jing, Y., Li, K., Zhang, Z., Zhang, J., Ma, Z., Yang, J., Zhang, L., Wu, Z., He, J., Xing, J., and Cheng, J. *Benchmarking the Limits of In-Context Reinforcement Learning for Ad-Hoc Teamwork*, 2026. https://arxiv.org/abs/2605.24423

Kazemkhani, S., Pandya, A., Cornelisse, D., Shacklett, B., and Vinitsky, E. *GPUDrive: Data-driven, multi-agent driving simulation at 1 million FPS*, 2025. https://arxiv.org/abs/2408.01584

Lupu, A., Cui, B., Hu, H., and Foerster, J. Trajectory Diversity for Zero-Shot Coordination. In M. Meila & T. Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139, pp. 7204–7213. PMLR, 2021. https://proceedings.mlr.press/v139/lupu21a.html

Ma, X., Li, J., Kochenderfer, M. J., Isele, D., and Fujimura, K. *Reinforcement Learning for Autonomous Driving with Latent State Inference and Spatial-Temporal Relationships*, 2021. https://arxiv.org/abs/2011.04251

McKee, K. R., Gemp, I., McWilliams, B., Duéñez-Guzmán, E. A., Hughes, E., and Leibo, J. Z. *Social diversity and social preferences in mixed-motive reinforcement learning*, 2020. https://arxiv.org/abs/2002.02325

Melo, L. C. *Transformers are Meta-Reinforcement Learners*, 2022. https://arxiv.org/abs/2206.06614

Mirsky, R., Carlucho, I., Rahman, A., Fosong, E., Macke, W., Sridharan, M., Stone, P., and Albrecht, S. V. *A Survey of Ad Hoc Teamwork Research*, 2022. https://arxiv.org/abs/2202.10450

Munkhdalai, T., and Yu, H. Meta networks. *International Conference on Machine Learning*, 2554–2563, 2017.

Rowe, L., Girgis, R., Gosselin, A., Carrez, B., Golemo, F., Heide, F., Paull, L., and Pal, C. *CtRL-Sim: Reactive and Controllable Driving Agents with Offline Reinforcement Learning*, 2024. https://arxiv.org/abs/2403.19918

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *Arxiv Preprint Arxiv:1707.06347*, 2017.

Strouse, D., McKee, K. R., Botvinick, M., Hughes, E., and Everett, R. *Collaborating with Humans without Human Data*, 2022. https://arxiv.org/abs/2110.08176

Suarez, J. *PufferLib: Making Reinforcement Learning Libraries and Environments Play Nice*, 2024. https://arxiv.org/abs/2406.12905

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. *International Conference on Machine Learning*, 9229–9248, 2020.

Tang, Z., Yu, C., Chen, B., Xu, H., Wang, X., Fang, F., Du, S., Wang, Y., and Wu, Y. *Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization*, 2021. https://arxiv.org/abs/2103.04564

Team, A. A., Bauer, J., Baumli, K., Baveja, S., Behbahani, F., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., and others. Human-timescale adaptation in an open-ended task space. *Arxiv Preprint Arxiv:2301.07608*, 2023.

Team, O. E. L., Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., McAleese, N., Bradley-Schmieg, N., Wong, N., Porcel, N., Raileanu, R., Hughes-Fitt, S., Dalibard, V., and Czarnecki, W. M. *Open-Ended Learning Leads to Generally Capable Agents*, 2021. https://arxiv.org/abs/2107.12808

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. *Attention Is All You Need*, 2017. https://arxiv.org/abs/1706.03762

Vinitsky, E., Lichtlé, N., Yang, X., Amos, B., and Foerster, J. *Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world*, 2023. https://arxiv.org/abs/2206.09889

Wang et al. *ROTATE: Regret-driven Open-ended Training for Ad Hoc Teamwork*, 2026. https://arxiv.org/abs/2505.23686

Wang et al. *CooT: Learning to Coordinate In-Context with Coordination Transformers*, 2026. https://arxiv.org/abs/2506.23549

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. *Learning to reinforcement learn*, 2017. https://arxiv.org/abs/1611.05763

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T.-Y. *On Layer Normalization in the Transformer Architecture*, 2020. https://arxiv.org/abs/2002.04745

Yu, C., Gao, J., Liu, W., Xu, B., Tang, H., Yang, J., Wang, Y., and Wu, Y. *Learning Zero-Shot Cooperation with Humans, Assuming Humans Are Biased*, 2023. https://arxiv.org/abs/2302.01605

Zhang, D., Xue, J., Cui, Y., Wang, Y., Liu, E., Jing, W., Chen, J., Xiong, R., and Wang, Y. *Zero-shot Transfer Learning of Driving Policy via Socially Adversarial Traffic Flow*, 2023. https://arxiv.org/abs/2304.12821

Zhao, R., Song, J., Yuan, Y., Haifeng, H., Gao, Y., Wu, Y., Sun, Z., and Wei, Y. *Maximum Entropy Population-Based Training for Zero-Shot Human-AI Coordination*, 2022. https://arxiv.org/abs/2112.11701

Zintgraf, L., Devlin, S., Ciosek, K., Whiteson, S., and Hofmann, K. *Deep Interactive Bayesian Reinforcement Learning via Meta-Learning*, 2022. https://arxiv.org/abs/2101.03864

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. *VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning*, 2020. https://arxiv.org/abs/1910.08348

Appendix

A Environment and Training Details

A.1 Observations and Actions

All positions and angles are expressed in the policy's own frame (Table 2). The ego observes every road user within 50 m and the road segments in a 21 × 21 grid of 5 m cells around it, nearest cells first. Observations are noise-free and not occluded. Actions are a single 91-way categorical choice over 7 accelerations $\{-4, -2.67, -1.33, 0, 1.33, 2.67, 4\}$ m/s² and 13 steering angles evenly spaced in $[-1, 1]$ rad, applied through a kinematic bicycle model with $\Delta t = 0.1$ s. Each trial lasts at most 201 steps.

Table 2: **Observation space.** Co-players additionally receive their conditioning vector z (5 values) appended to their own-state features.

Group	Features
Own state (9)	goal position (x, y); signed speed; width; length; collision flag; lateral offset from the lane centre; cosine of the heading relative to the lane; respawn flag (always 0 in our setting)
Road users (up to 63 × 7)	relative position (x, y); relative heading $(\cos, \sin)$; width; length; signed speed. All vehicles, cyclists and pedestrians within 50 m; unused slots are zero
Road segments (up to 200 × 7)	midpoint (x, y); direction $(\cos, \sin)$; half-length; width (constant); type (lane centre, road line, road edge, stop sign, crosswalk, speed bump, driveway)

A.2 Rewards

The ego receives +1 for reaching its goal (within 2 m), −0.5 for each step in collision with another vehicle and −0.5 for each step off-road. Collisions and off-road events do not end the trial, so these penalties can recur on consecutive steps. An ego that reaches its goal is removed from the scene until the next trial. Both the ego and the co-players also receive a lane-alignment term,

$$r_t^{\text{lane}} = \lambda \Delta t[\min(\cos\theta_t, 0) + \min(v_t \cos\theta_t, 0) + 0.0025(1 - |\theta_t| \,/(\pi/2))], \tag{4}$$

where θ_t is the heading error relative to the nearest lane and v_t is the speed. It is close to zero when driving along the lane and penalizes driving against the lane direction in proportion to speed. We use $\lambda = 0.05$ for the ego and $\lambda = 0.01$ for the co-players.

A.3 Architecture and Optimization

Each observation group is encoded by two linear layers with layer normalization into 128 dimensions. Road users and road segments are encoded per object with shared weights and max-pooled, and the road-segment type is one-hot encoded. The three encodings pass

through a GELU and a linear projection to a 256-dimensional embedding. The transformer has two pre-LN layers with 4 attention heads, model width 256, feed-forward width 512, GELU activations, fixed sinusoidal positional encodings and a final layer normalization, followed by linear policy and value heads. Training hyperparameters are listed in Table 3. Value targets and advantages are computed within each trial, with bootstrapping stopped at trial boundaries. Each ego run used a single NVIDIA H100 GPU and took about 33 hours for $k = 4$.

Table 3: **Ego training hyperparameters**, shared across all configurations.

Hyperparameter	**Value**
Algorithm	PPO with prioritized minibatch sampling ($\alpha = \beta_0 = 0.85$) and V-trace clipping ($\overline{\rho} = \overline{c} = 1$)
Learning rate	3×10^{-3}, linearly annealed
Discount γ / GAE λ	0.995 / 0.95
Clip coefficient / value clip	0.2 / 0.2
Value coefficient / entropy coefficient	2.0 / 0.005
Max gradient norm / update epochs	1.0 / 1
Training scenes	4,504 nuPlan scenes
Evaluation scenes	539 nuPlan scenes
Environment steps	3×10^9, counted over all vehicle slots

A.4 Co-player Population

The co-player uses the same architecture as the ego, with a context of one trial (201 steps) and the conditioning vector z appended to its own-state features. For each entropy bound $e \in \{0.05, 0.10, 0.20, 0.50\}$, we train one co-player network with PPO for 2×10^9 environment steps on nuPlan scenes, sampling z per scene from the ranges in Table 1. The sampled discount γ is used in the advantage estimates and the sampled coefficient β weights the entropy bonus. Training one co-player takes about 5 hours on a single GPU. Because each entropy bound uses its own pretrained network, comparisons across entropy bounds also compare different co-player networks. Co-players do not appear during evaluation, where all other vehicles replay human logs.

B Evaluation Protocol and Scene Selection

Scenes. The evaluation pool consists of the 10% of nuPlan scenes with the most interaction involving the recorded ego vehicle, of which 539 are valid scenes. The ego's training pool of 4,504 scenes shares no overlap with it. The two pools are drawn from the same cities, so different clips of the same road locations can appear in both.

Rollouts. In each scene, one vehicle is controlled by the ego policy and all others replay their recorded trajectories, which restart at every trial boundary. We run 20 rollouts per scene with sampled actions for each of the three training seeds. For the memory-reset control, the ego's context and positional index are cleared at every trial boundary, so every trial is played as a first trial. The control is evaluated on the same scenes as the intact policy.

Adaptable scenes. For each trained policy, a scene is adaptable if its first-trial success rate over the 20 rollouts is below 0.8.

Results without scene selection. Selecting scenes with a poor first trial inflates ΔR, because part of a poor first-trial result is chance. Figure 7 therefore reports the per-trial 10th percentile and mean of the return over all 539 scenes, with no selection. The adaptation remains visible for $e = \{0.05, 0.10, 0.20\}$, but for $e = 0.50$, return does not improve without selection.

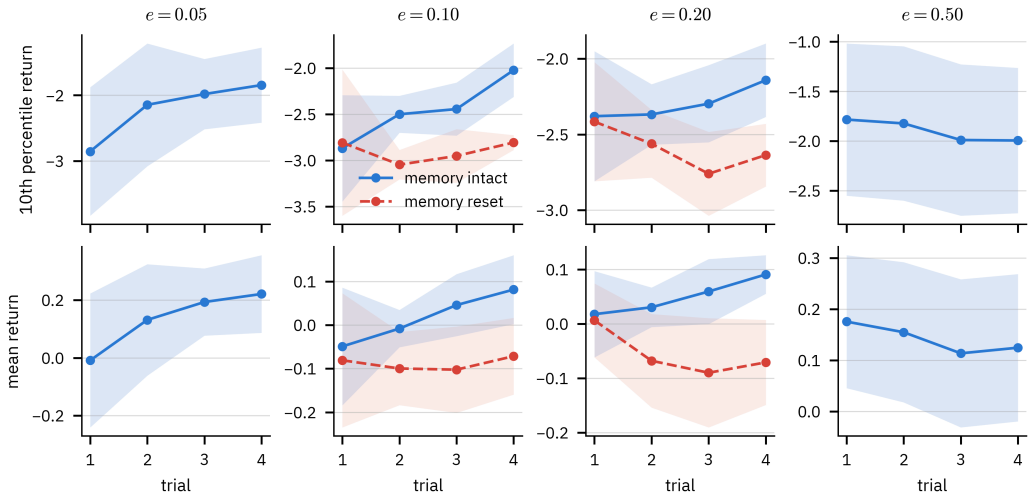


Figure 7: **Without scene selection, adaptation remains for low-entropy co-players and disappears at $e = 0.50$.** Per-trial 10th percentile (top) and mean (bottom) of the return over all 539 held-out scenes, with memory intact (solid) and reset at every trial boundary (dashed). Mean over three training seeds, each evaluated with 20 rollouts per scene; shading ± 1 SD across seeds.

C Additional Results

Number of trials. Figure 8 shows per-trial return for $k = 2$ to 5 at $e = 0.10$. Return improves across trials at every episode length.

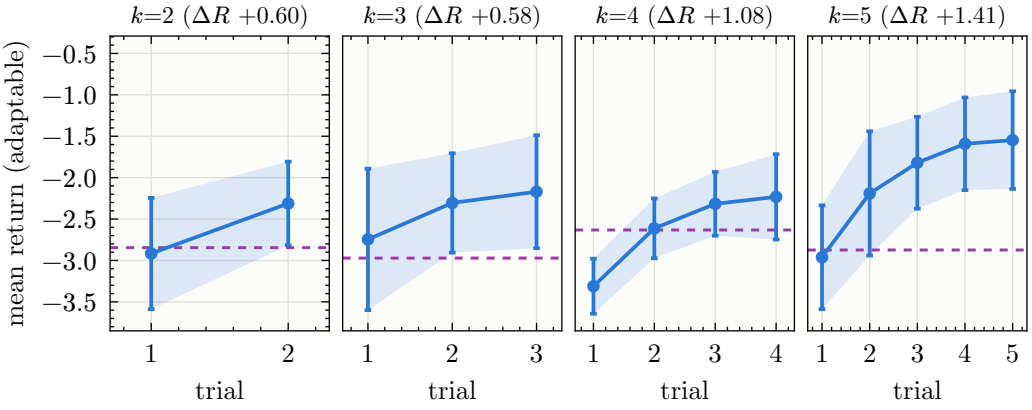


Figure 8: Per-trial return at every episode length

Co-player entropy. Figure 9 shows that ΔR falls steadily as the co-player entropy bound grows, from +1.79 at $e = 0.05$ to +0.55 at $e = 0.50$. First-trial success over all scenes stays between 0.89 and 0.92, with differences within seed variation.

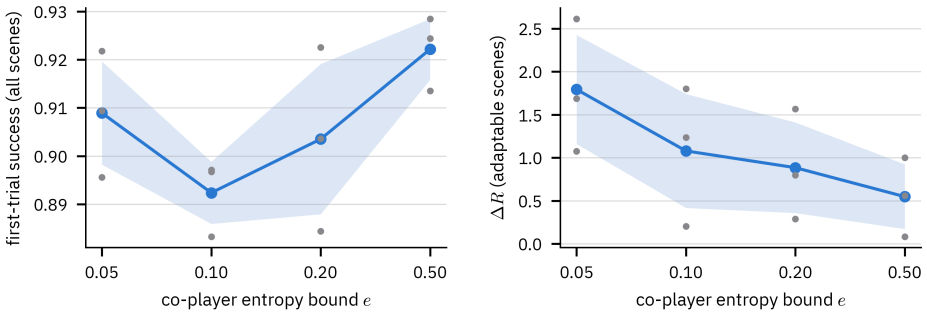


Figure 9: **Adaptation falls as co-player entropy grows, while first-trial success changes little.** Left: first-trial success over all 539 scenes. Right: ΔR on adaptable scenes. Line and shading show the mean ± 1 SD over three training seeds, and dots show individual seeds.

Model width. Figure 10 varies the width of the encoder output and transformer ($d \in \{64, 128, 256\}$) at $e = 0.10$. The widest model adapts the most ($+1.08$ against $+0.54$ and $+0.60$), and first-trial success also increases slightly with width.

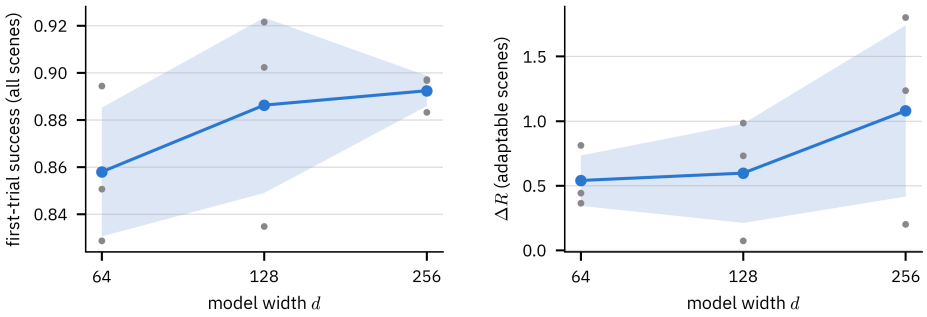


Figure 10: **Wider models adapt more.** First-trial success over all scenes (left) and ΔR on adaptable scenes (right) for model widths $d \in \{64, 128, 256\}$, with $e = 0.10$ and $k = 4$. Mean ± 1 SD over three training seeds; dots show individual seeds.

D Prioritized Adaptation Replay

We test whether prioritizing training scenes by their adaptation gap improves adaptation, following prioritized level replay (PLR) (Jiang et al., 2021).

Setup. Let $\mathcal{M}$ be the set of training scenes. Each episode samples a scene m from a distribution P over $\mathcal{M}$, and uniform sampling sets $P(m) = 1/\,|\mathcal{M}|$. Within an episode of k trials, let $S_i(m, \theta) \in [0, 1]$ denote the ego's expected success in trial i on scene m, where the policy in trial i conditions on its memory of trials $1, ..., i-1$.

Regret. Let $V^*(m)$ be the best success achievable on m by any policy. The regret of the first trial is

$$\text{Regret}(m; \theta) = V^*(m) - S_1(m, \theta). \tag{5}$$

Since V^* is unobservable, we compare two proxies for it.

1. **Difficulty (PLR).** Using the trivial bound $V^*(m) \leq 1$ gives the score $f_{\text{diff}}(m; \theta) = 1 - S_1(m, \theta)$.

2. **Adaptation gap (PAR, ours).** Using the agent's own memory-informed last trial as the reference gives $f_{\text{gap}}(m;\theta) = S_k(m,\theta) - S_1(m,\theta)$. Since the last-trial policy is itself a policy, $S_k(m,\theta) \le V^*(m)$, and therefore $f_{\text{gap}}(m;\theta) \le \text{Regret}(m;\theta)$ for every scene. The adaptation gap is thus a conservative estimate of regret that is large only where the agent can already improve through experience.

Implementation. Both scores are estimated online from training rollouts. For each scene, we track an exponential moving average (rate 0.05) of the mean first-trial success and of the mean last-minus-first success gap over the egos in each episode, clipping the gap at zero. Scenes not yet visited receive an optimistic score of 0.5. Scenes are sampled with probability

$$P(m) = \frac{\lambda}{|\mathcal{M}|} + (1-\lambda)\frac{f(m)}{\sum_{m'} f(m')} \tag{6}$$

with a uniform mixing weight $\lambda = 0.2$.

Results. Figure 11 compares uniform sampling, PLR by difficulty and PAR, with three seeds each ($e = 0.10$, $k = 4$). Seed-to-seed variation within each arm exceeds the differences between arms, and we observe no reliable effect of the curriculum at this scale.

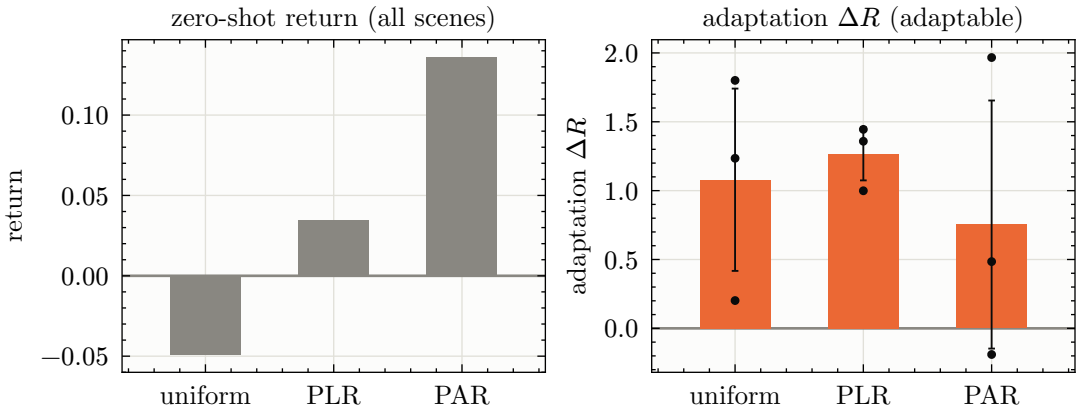


Figure 11: No curriculum effect at this scale. Uniform sampling, PLR by difficulty, and PAR by adaptation gap (ours), 3 seeds each. Seed spread within an arm exceeds the gaps between arms — PAR's seeds span −0.19 to 1.97.