Z2SP: TURNING ZERO-SHOT COORDINATION INTO SELF-PLAY VIA PARTNER-SPECIFIC TEST-TIME ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot coordination (ZSC) requires independently trained agents to coordinate with unseen partners in cooperative multi-agent reinforcement learning (MARL). Most existing methods learn policies that generalize broadly before deployment. Once a particular partner appears, however, the optimization focus of ZSC shifts from distribution-level generalization to online partner-specific policy estimation and adaptation. When the partner's self-play return exceeds that of the initial ZSC pairing, the current partner policy provides a concrete adaptation target: moving the ego policy toward it can steer coordination toward a self-play-like mode, narrowing the ZSC-to-SP gap and improving ZSC performance. We present **Z2SP**, which constructs a finite-sample approximation of the inaccessible partner policy over encountered observations from online partner-view observation–action samples and directly adapts the ego actor toward this target. This unifies partner-policy estimation and ego-policy adaptation without a separate partner model. All test-time adaptation uses samples collected after the initial zero-shot encounter. For persistent adaptation, Z2SP retains its adaptation state across episodes, reuses accumulated samples through delayed history replay, and employs closed-loop meta-initialization to mitigate early adaptation cost. Across five OvercookedV2 layouts with 20 unseen partners from four MARL families, Z2SP reaches 91.8% of partner self-play return after 40 episodes of persistent adaptation and 98.0% over the final 20 episodes, averaged across layouts.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) studies how to learn policies that enable effective coordination with partners in shared tasks. Such coordination is critical for multi-robot systems (Agrawal et al., 2023), cooperative games (Gessler et al., 2025), and human–AI collaboration (Carroll et al., 2019). Since future deployment partners are generally unavailable during training, self-play (SP) is a prevalent mechanism for generating cooperative experience. In SP, agents governed by the same policy repeatedly interact to optimize shared team return (Hu et al., 2020; Gessler et al., 2025).

Although SP-trained agents can achieve strong task performance, independent training runs and methods may develop incompatible self-consistent coordination conventions. This incompatibility is a central difficulty in zero-shot coordination (ZSC), which studies how independently trained agents coordinate with partners unseen during training (Hu et al., 2020). Convention mismatch can make the initial ZSC return substantially lower than the partner's SP return. We call this difference the *ZSC-to-SP gap*.

A dominant line of ZSC research addresses this mismatch before deployment by learning policies that generalize across broad partner spaces. Representative approaches remove conventions induced by arbitrary symmetry breaking (Hu et al., 2020), expose agents to diverse partner populations or trajectories (Lupu et al., 2021; Strouse et al., 2021; Zhao et al., 2023), or broaden the coverage and scale of the training distribution (Yan et al., 2023; Hui et al., 2026). Such generalization is necessary while the target partner remains unknown, but a finite training distribution cannot cover every possible partner. Once a particular partner appears and begins generating interaction data,

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.

both the available information and the optimization focus change: distribution-level generalization over unknown partners can shift to online partner-specific policy estimation and adaptation. Pre-deployment generalization remains a valuable starting point, but does not itself perform this partner-specific parameter adaptation. This raises a key question: toward what target should the ego policy be adapted?

The ZSC-to-SP gap provides a natural answer. The partner's SP return comes from same-policy coordination in which both agents follow the partner policy. When policy incompatibility causes the initial ZSC pairing to underperform this SP pairing, adapting the ego policy toward the partner policy over coordination-relevant observations can move the pairing toward the self-play-like mode familiar to the partner, thereby narrowing the ZSC-to-SP gap and improving ZSC performance. The current partner policy therefore provides a concrete adaptation target. At test time, however, the ego agent can access neither the partner's policy parameters nor its conditional action distributions. We address this challenge by accumulating partner-view observation–action samples during interaction and using them to construct a finite-sample approximation of the partner policy over the encountered observations.

Based on this insight, we propose **Z2SP**, a test-time adaptation framework that transforms the ego policy toward a finite-sample estimate of the current partner policy. In the fully observable tasks considered here, Z2SP uses the partner observation–action pair available after each joint transition, without access to policy parameters or future information. These samples supervise the ego actor directly, unifying partner-policy estimation and ego-policy adaptation without a separate partner model. For persistent adaptation, it carries the adapted policy, Adam state, and replay buffer across episodes, uses delayed history replay to reuse accumulated samples, and employs a meta-learned initialization that mitigates early adaptation cost.

We evaluate Z2SP across five OvercookedV2 layouts with 20 unseen partners from four MARL algorithm families under both single-episode adaptation and persistent adaptation to the same unseen partner across episodes. Under persistent adaptation, Z2SP reaches 91.8% of partner self-play return after 40 episodes and 98.0% over the final 20 episodes of the 100-episode horizon, averaged across layouts. Further experiments analyze continual re-adaptation after partner switches, role coverage, delayed history replay, meta-initialization, and online computational cost.

Our contributions are threefold:

- We identify a deployment-time shift in ZSC from distribution-level generalization to online partner-specific estimation and adaptation, and propose the current partner policy as the target for online adaptation. Adapting the ego policy toward this target can move ZSC toward self-play-like coordination, narrowing the ZSC-to-SP gap and improving ZSC performance.
- We propose Z2SP, which directly adapts the ego actor using a finite-sample objective over online partner-view observation–action samples, approximating the inaccessible partner policy without a partner model. We further introduce cross-episode state retention and delayed history replay for persistent adaptation, together with closed-loop meta-initialization to mitigate early adaptation cost.
- We validate Z2SP across five layouts and 20 unseen partners from four MARL families, demonstrating progressive improvement toward partner self-play and analyzing partner switches, role coverage, ablations, and online efficiency.

2 Related Work

ZSC relates to ad hoc teamwork with unfamiliar teammates (Stone et al., 2010). Methods emphasize pre-deployment generalization or online adaptation.

2.1 Generalization of Fixed Policies

Other-Play addresses arbitrary symmetry breaking (Hu et al., 2020), while Off-Belief Learning controls belief-based reasoning to avoid incompatible conventions (Hu et al., 2021). Population-based methods diversify partners or trajectories: TrajeDi promotes trajectory diversity, Fictitious Co-Play

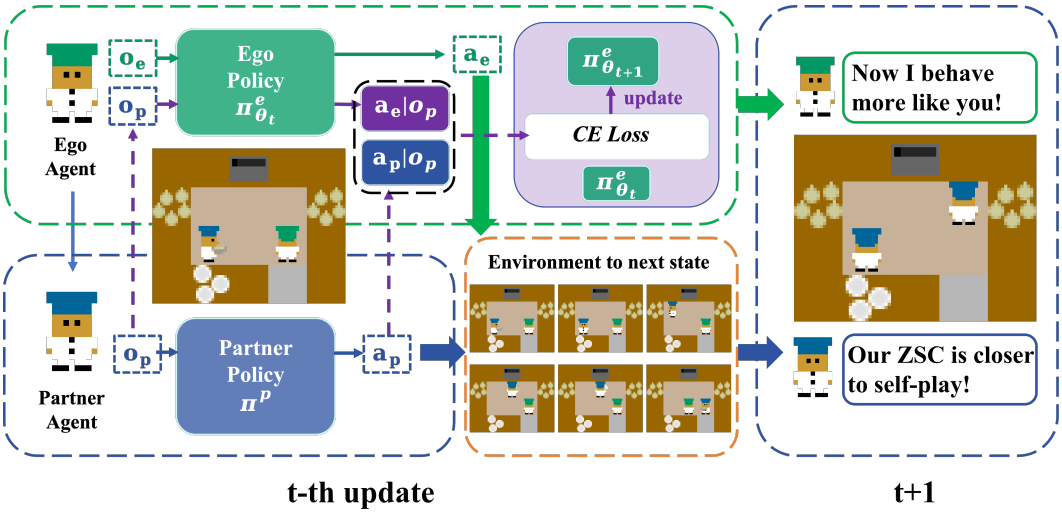


Figure 1: Overview of Z2SP: improving ZSC by approaching the partner's self-play mode.

samples training stages and seeds, and MEP maximizes population entropy (Lupu et al., 2021; Strouse et al., 2021; Zhao et al., 2023). COLE uses open-ended population objectives (Li et al., 2023); E3T and ScaPT improve partner generation and training coverage (Yan et al., 2023; Hui et al., 2026). GAMMA models partner strategies generatively (Liang et al., 2024), while Cross-Environment Cooperation diversifies coordination tasks (Jha et al., 2025).

2.2 Online Adaptation to Unseen Partners

Strong initial ZSC performance does not necessarily imply efficient few-shot adaptation (Nekoei et al., 2023). LIAM infers agent representations from local interaction history at execution (Papoudakis et al., 2021a), while TALENTS selects responses through online latent-strategy inference (Li et al., 2025). CooT adapts in context from recent interaction histories without test-time parameter updates (Wang et al., 2026). More broadly, TTT and Tent update model parameters at test time using self-supervision or entropy minimization (Sun et al., 2020; Wang et al., 2021a). Z2SP instead uses partner-view observation–action samples as supervision, directly updating the ego actor across episodes. This enables partner-specific optimization without a separate partner model or selection among fixed responses.

3 Method

We consider fully observable cooperative tasks (Carroll et al., 2019; Papoudakis et al., 2021b) and use partner observation–action histories available during interaction. We focus on an ego–partner pair, with the unseen partner's policy fixed within each adaptation stage.

Motivated by the observation that policies can perform well in self-play yet coordinate poorly with unseen partners, we propose **Z2SP**, a test-time adaptation framework. Adapting the ego policy toward the current partner's policy can improve ZSC by approaching the partner's self-play coordination mode. Figure 1 illustrates its online adaptation mechanism.

3.1 Online Ego-to-Partner Policy Adaptation

Let θ_t denote the ego actor parameters before the update at step t, and let $\pi^e_{\theta_t}$ and π^p denote the ego and partner policies, respectively. We write $J(\pi^e, \pi^p)$ for their expected episodic team return. Ideally, the adapted ego policy approximates the partner policy on an ego-view observation o^e encountered during coordination:

$$\pi^e_{\theta_t}(\cdot \mid o^e) \approx \pi^p(\cdot \mid o^e). \tag{1}$$

The ZSC pairing then approaches behavioral self-play:

$$J(\pi^e_{\theta_t}, \pi^p) \approx J(\pi^p, \pi^p). \tag{2}$$

Therefore, when the partner's self-play performance exceeds its initial ZSC performance with the ego agent, transforming the ego policy toward the partner policy provides a direct route to improved coordination.

At test time, the partner's policy parameters and conditional action distributions are not directly accessible. We address this challenge by constructing a finite-sample estimate of its policy as an observable adaptation target. The policy remains fixed within each adaptation stage, while interaction continually supplies partner-view observation–action samples for estimating it.

After the joint transition at step t, Z2SP pairs the partner's executed action $a^p_t \sim \pi^p(\cdot \mid o^p_t)$ with its corresponding environment-provided observation o^p_t for subsequent policy updates. Each pair provides one sample from the partner's conditional action distribution. After observing $t+1$ such samples, we define the finite-sample behavioral estimate of π^p as

$$\widehat{\pi}^p_{t+1} \in \arg\max_{\pi \in \mathcal{P}} \frac{1}{t+1} \sum_{\tau=0}^{t} \log \pi\left(a^p_\tau \mid o^p_\tau\right), \tag{3}$$

where $\mathcal{P}$ denotes the policy class representable by the ego actor. This maximum-likelihood objective selects the representable policy that best explains the partner actions observed so far, yielding an estimate of π^p over the covered observation support. With sufficient behavioral coverage and adequate function approximation, $\widehat{\pi}^p_{t+1}$ can more accurately approximate π^p over the interaction-relevant observations.

Rather than recomputing the full finite-sample objective in Eq. (3) after every transition, Z2SP approximates it online by applying each arriving partner-view sample directly to the ego actor. With compatible observation encodings, we feed o^p_t into the ego actor and use a^p_t as the label to compute categorical cross-entropy over the common discrete action space $\mathcal{A}$:

$$\mathcal{L}^{\text{in}}_t(\theta_t) = -\log \pi^e_{\theta_t}\left(a^p_t \mid o^p_t\right). \tag{4}$$

At each step, we perform one gradient-based update to the ego actor using $\mathcal{L}^{\text{in}}_t$, while keeping the critic frozen. Ego actions remain conditioned on ego-view observations.

Within each episode, Z2SP carries the policy parameters and Adam (Kingma & Ba, 2015) state from one step to the next. Let m index episodes of horizon T, and let $(\theta_{m,t}, z_{m,t})$ denote the adaptation state after t interaction steps in episode m. For persistent adaptation to the same partner, we retain $(\theta_{m+1,0}, z_{m+1,0}) = (\theta_{m,T}, z_{m,T})$ across episodes while resetting only the environment state.

3.2 Delayed History Replay for Persistent Adaptation

Latest-sample updates are inexpensive but underuse accumulated partner information and are sensitive to individual samples. We introduce delayed history replay to reuse this information and stabilize adaptation.

Let $\mathcal{R}_{m,t}$ denote the buffer contents immediately before processing $(o^p_{m,t}, a^p_{m,t})$, where $t \in \{0, \ldots, T-1\}$. Initially empty and retained across episodes, the buffer stores at most the B most recent completed partner samples.

After receiving $(o^p_{m,t}, a^p_{m,t})$, we perform the standard online update. Once the replay delay has elapsed and sufficient history is available, we form $\mathcal{S}_{m,t} \subseteq \mathcal{R}_{m,t}$ from m_o approximately uniformly spaced older samples and m_r most recent samples, with $|\mathcal{S}_{m,t}| = m_o + m_r$, then perform one additional Adam update from the updated policy and optimizer state using

$$\mathcal{L}^{\text{rep}}_{m,t}(\theta) = \beta\big[-\log \pi^e_\theta(a^p_{m,t} \mid o^p_{m,t})\big] + \frac{1-\beta}{|\mathcal{S}_{m,t}|} \sum_{(o,a)\in\mathcal{S}_{m,t}} \big[-\log \pi^e_\theta(a \mid o)\big]. \tag{5}$$

Here, $\beta \in [0,1]$ balances current and historical samples. The resulting policy and Adam state are $(\theta_{m,t+1}, z_{m,t+1})$. We then append the current sample to the FIFO buffer, removing the oldest entry if needed.

Our default persistent-adaptation configuration uses $B = 256$, $m_o = 4$, $m_r = 3$, and $\beta = 0.5$. Replay begins after two complete episodes. This delay makes the first two episodes identical to standard Z2SP adaptation and avoids repeatedly amplifying noisy early evidence.

3.3 Closed-Loop Sequential Meta-Initialization

During early adaptation, the ego policy may not consistently follow either its original convention or the partner's, causing a performance drop. To reduce this transition cost, we learn an adaptation-aware initialization θ_0 through bilevel meta-learning Finn et al. (2017), with partner-view adaptation at the inner level and ego-view behavioral evaluation at the outer level. We construct the auxiliary population $\Pi_{\text{meta}} = \{\pi_j^{\text{pop}}\}_{j=1}^{M}$ from self-play-trained agents disjoint from the test-time partners.

Inner-level adaptation. For each sampled population partner, we collect temporally ordered support–query pairs $((o_t^p, a_t^p), o_{t+1}^e)$. Starting each sequence from θ_0 with a fresh Adam state, we apply the test-time online update sequentially to its support samples, yielding $\theta_{j,t+1}$ after processing (o_t^p, a_t^p).

Outer-level evaluation. The outer level evaluates the adapted policy on the ego-view query observation. Since the population-partner checkpoints are available offline, we construct the target

$$y_{j,t+1} = \arg\max_{a \in \mathcal{A}} \pi_j^{\text{pop}}(a \mid o_{t+1}^e), \tag{6}$$

which specifies how population partner j would act from the ego agent's current view. Let θ_{base} denote the actor parameters of the base ego policy. The query loss is

$$\mathcal{L}_{j,t}^{\text{qry}} = -\log \pi_{\theta_{j,t+1}}^e(y_{j,t+1} \mid o_{t+1}^e) + \lambda_{\text{ret}} D_{\text{KL}}\Big(\pi_{\theta_{\text{base}}}^e(\cdot \mid o_{t+1}^e) \,\|\, \pi_{\theta_{j,t+1}}^e(\cdot \mid o_{t+1}^e)\Big). \tag{7}$$

The first term encourages partner-view updates to produce the desired behavior on ego-view observations, while the second retains useful behavior inherited from the base policy.

We also regularize the unadapted initialization using

$$\mathcal{L}_j^{\text{init}} = \mathbb{E}_{o^e \sim \mathcal{D}_j}\left[D_{\text{KL}}\big(\pi_{\theta_{\text{base}}}^e(\cdot \mid o^e) \,\|\, \pi_{\theta_0}^e(\cdot \mid o^e)\big)\right]. \tag{8}$$

where $\mathcal{D}_j$ is the empirical distribution of ego-view query observations collected with population partner j. The outer objective is

$$\min_{\theta_0} \quad \mathbb{E}_j\left[\frac{1}{|\mathcal{Q}_j|}\sum_{t \in \mathcal{Q}_j} \mathcal{L}_{j,t}^{\text{qry}} + \lambda_{\text{init}} \mathcal{L}_j^{\text{init}}\right] + \frac{\lambda_{\text{anc}}}{2}\|\theta_0 - \theta_{\text{base}}\|_2^2, \tag{9}$$

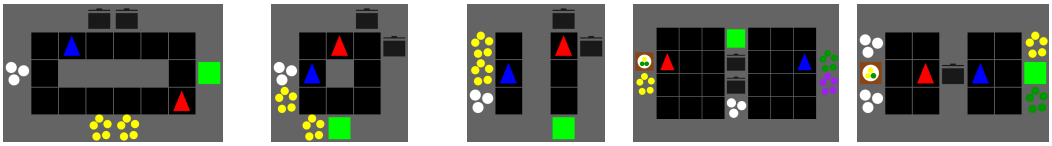
where $\mathcal{Q}_j$ contains the query steps used for meta-training and λ_{ret}, λ_{init}, λ_{anc} are nonnegative regularization weights. We optimize this objective using first-order meta-gradients Nichol et al. (2018) and keep the critic frozen.

Closed-loop recollection. Because sequential adaptation changes the states subsequently visited, we recollect closed-loop trajectories after the first meta-training round and repeat meta-training once using the resulting support–query sequences. Population-partner checkpoints and ego-view targets are used only during offline meta-training.

4 Experiments

4.1 Settings

Environment and partners. We evaluate Z2SP on five OvercookedV2 layouts: Counter Circuit, Coordination Ring, Forced Coordination, More Fun Coordination, and Fun Symmetries Plates (Mixed). Figure 2 visualizes their initial configurations. Each episode lasts at most 400 steps and uses a shared team reward. The evaluation set contains 20 self-play-trained partners unseen during meta-initialization, with five policies from each of IQL (Tan, 1993), VDN (Sunehag et al., 2018), MAAC (Iqbal & Sha, 2019), and HAPPO (Kuba et al., 2022).



(a) Counter Circuit **(b) Coordination Ring** **(c) Forced Coordination** **(d) More Fun Coordination** **(e) Sym. Plates (Mixed)**

Figure 2: The five OvercookedV2 layouts used in our main evaluation. The red and blue triangles indicate the two agents at their initial positions. For recipe-dependent layouts, the visualization shows one valid initial recipe.

Baselines. We consider three baseline categories: self-play methods, including PPO (Schulman et al., 2017) and PPO-SA (Gessler et al., 2025); population-based methods, including MEP (Zhao et al., 2023) and TrajeDi (Lupu et al., 2021); and TALENTS (Li et al., 2025), which adapts its latent-strategy estimate online without updating policy parameters. Z2SP uses a PPO policy trained under the same budget as its base policy.

Implementation details. We train each comparison method for 10 million environment steps with five seeds derived from 42. Z2SP uses OvercookedV2's official CNN-based PPO baseline. The disjoint meta-training population comprises self-play-trained MAPPO (Yu et al., 2022), WQMIX (Rashid et al., 2020), and QPLEX (Wang et al., 2021b). Meta-initialization uses two closed-loop rounds of 300 outer updates. Test-time Adam uses learning rate 1.7919×10^{-5} and global gradient clipping at 10. For meta-initialization, we set $\lambda_{\text{ret}} = 1$, $\lambda_{\text{init}} = 10$, and $\lambda_{\text{anc}} = 10^{-4}$, and keep them fixed across both closed-loop meta-training rounds.

Metrics. Unless otherwise stated, reported returns are averaged over all runs of the method, the relevant partner set, both deployment roles, and 20 evaluation seeds per partner. For layout ℓ, SP-normalized return after m adaptation episodes is $S_{\ell,m} = 100R_{\ell,m}/R_{\ell,\text{SP}}$, where $R_{\ell,m}$ and $R_{\ell,\text{SP}}$ denote the adapted-policy and partner self-play returns, respectively. Overall averages returns and SP-normalized percentages separately across the five layouts.

Table 1: Baseline comparison and single-episode adaptation across five layouts. Entries report mean return ± 95% confidence-interval half-width. E_0 and E_1 denote evaluation before and after one adaptation episode. Bold marks the highest mean return in each column.

		Five evaluation layouts and their overall average					
Method	Base model	Counter	Coord. Ring	Forced	More Fun	Sym. Plates	**Overall**
PPO(baseline)	–	74.4±23.3	102.8±34.1	21.3±19.7	65.4±27.9	79.4±33.0	68.6±10.5
PPO-SA	PPO	60.5±17.3	94.7±25.2	48.5±13.5	**131.3**±27.4	127.6±30.6	92.5±11.0
MEP	PPO	58.8±14.3	88.8±25.1	53.8±18.4	82.1±10.3	97.6±8.2	76.2±9.1
TrajeDi	PPO	56.6±9.9	91.3±23.2	39.6±13.7	105.6±16.4	88.0±21.4	76.2±8.3
TALENTS	PPO	**78.4**±24.4	**112.4**±27.3	46.5±15.7	127.3±25.8	**129.0**±27.1	**98.7**±10.3
Z2SP (E_0)	PPO	75.2±22.6	108.0±30.6	35.3±20.0	66.0±25.3	77.8±30.9	72.4±9.4
Z2SP (E_1)	PPO	77.5±23.5	111.0±31.7	**67.5**±18.1	68.7±24.7	70.4±25.8	79.0±9.0

4.2 Baseline Comparison and Single-Episode Adaptation

We compare the baselines with Z2SP before adaptation (E_0) and after one episode of partner-specific adaptation (E_1). Z2SP is evaluated in separate rollouts with its policy parameters held fixed. Table 1 shows that TALENTS achieves the highest overall mean return of 98.7 and performs best on Counter Circuit, Coordination Ring, and Fun Symmetries Plates. PPO-SA performs best on More Fun Coordination, whereas Z2SP at E_1 performs best on Forced Coordination. MEP and TrajeDi obtain the same overall return of 76.2 but exhibit different layout-specific behavior. Thus, no method dominates across all layouts in this comparison.

Within Z2SP, the first adaptation episode raises the average return from 72.4 at E_0 to 79.0 at E_1. The effect remains layout dependent: Forced Coordination improves from 35.3 to 67.5, whereas Fun Symmetries Plates decreases from 77.8 to 70.4. These results show that one episode can provide useful partner evidence, but is not yet sufficient for consistent specialization across layouts.

4.3 Closing the ZSC-to-SP Gap through Persistent Adaptation

Table 2: Persistent adaptation toward partner self-play across five layouts. Each cell reports mean return ± 95% confidence-interval half-width, with SP-normalized return (%) below. E_m evaluates the policy after m adaptation episodes; Final 20 averages E_{81}–E_{100}. Partner SP is the reference. Bold marks the highest mean adaptation return in each column.

	Return ± 95% CI half-width / SP-normalized return (%)					
Stage	Counter	Coord. Ring	Forced	More Fun	Sym. Plates	**Overall**
Partner SP	164.9±8.0 100%	233.9±18.5 100%	199.7±2.6 100%	186.1±29.9 100%	196.5±9.1 100%	196.2±7.1 100%
Z2SP (E_0)	75.2±22.6 45.6%	108.0±30.6 46.2%	35.3±20.0 17.7%	66.0±25.3 35.5%	77.8±30.9 39.6%	72.4±9.4 36.9%
Z2SP (E_{10})	107.4±21.2 65.1%	166.1±18.9 71.0%	117.3±24.5 58.7%	78.2±23.7 42.0%	141.2±22.2 71.9%	122.0±7.9 61.8%
Z2SP (E_{20})	135.5±16.9 82.2%	206.2±21.1 88.2%	174.4±17.1 87.3%	101.1±21.3 54.3%	177.9±17.3 90.6%	159.0±9.1 80.5%
Z2SP (E_{40})	157.6±10.3 95.6%	222.3±17.2 95.1%	189.6±9.2 95.0%	139.9±23.5 75.2%	193.3±9.1 98.4%	180.5±5.7 91.8%
Z2SP (Final 20)	**162.5**±8.5 98.6%	**228.8**±18.2 97.8%	**196.2**±3.0 98.3%	**178.7**±25.9 96.0%	**194.6**±9.0 99.1%	**192.2**±6.2 98.0%

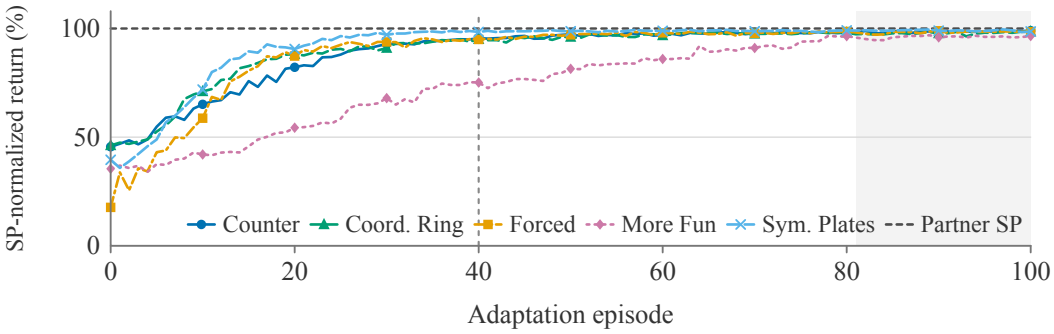


Figure 3: SP-normalized return during persistent adaptation across five OvercookedV2 layouts; 100% denotes the partner SP reference. The vertical dotted line marks E_{40}, and shading marks the final 20 episodes (E_{81}–E_{100}).

We evaluate persistent adaptation to the same unseen partner over a 100-episode horizon. Across episodes, Z2SP retains its policy parameters, Adam state, and replay buffer while alternating roles. Table 2 and Figure 3 report performance throughout adaptation.

By E_{40}, Z2SP reaches an average return of 180.5 and an average SP-normalized return of 91.8%. It exceeds 90% of partner SP return on four layouts, while More Fun Coordination adapts more slowly, reaching 75.2%. Over the final 20 episodes, the average return further increases to 192.2, approaching the partner self-play return of 196.2. Average SP-normalized return reaches 98.0%, ranging from 96.0% to 99.1% across the five layouts. In particular, More Fun Coordination reaches 96.0% over the final 20 episodes, indicating slower adaptation rather than a lower eventual coordination level.

The two protocols characterize Z2SP's short-term performance and persistent adaptation capability, respectively. In the baseline comparison, Z2SP at E_1 remains 19.7 points below TALENTS and

13.5 points below PPO-SA, whereas persistent adaptation raises Z2SP's own return from 79.0 at E_1 to 180.5 at E_{40} and 192.2 over the final 20 episodes. The methods differ in how they use partner information: MEP and TrajeDi improve pre-deployment generalization through population or trajectory diversity but keep their deployed policies fixed; TALENTS infers the partner's strategy online within an offline-learned latent strategy space and adjusts its conditioned response without updating policy-network parameters. In contrast, Z2SP continually updates the ego policy using accumulated partner-view observation–action samples. The results show that these continual online updates yield substantial long-term performance gains, progressively narrowing the ZSC-to-SP gap and bringing coordination performance close to the partner's self-play level.

4.4 Further Analysis and Ablations

Re-adaptation under partner switches. We test partner specificity and re-adaptation on Counter Circuit using a single Z2SP policy in an $A \to B \to A$ sequence, with 50 episodes per stage and no reset of the actor, Adam state, or replay buffer. Figure 4 shows sharp drops after both switches, indicating partner-specific specialization. Yet continued updates from new partner-view samples retarget the ego policy and restore near-self-play coordination after a temporary adaptation period, showing that this specialization is reversible. Across the three stages, Z2SP reaches 99.12%, 95.00%, and 98.80% of the corresponding partner SP returns, first exceeding 90% after 28, 29, and 9 episodes.

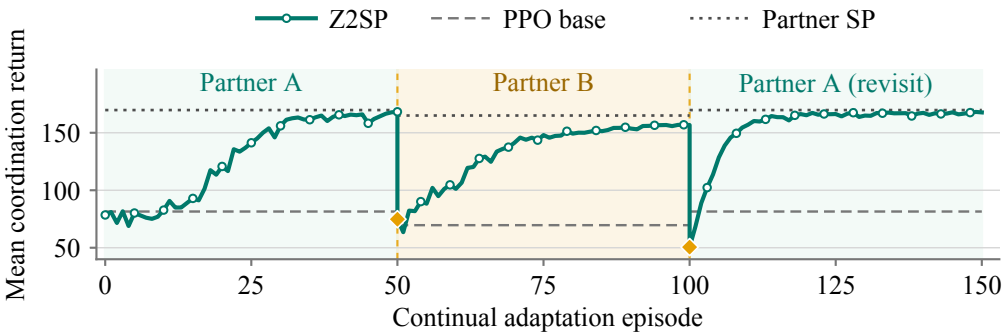


Figure 4: Partner switch test on Counter Circuit.

Role Coverage during Persistent Adaptation Figure 5 compares alternating- and fixed-role adaptation on four layouts. Fixed-role adaptation improves coordination on all four and approaches alternating-role performance on Counter Circuit and Coordination Ring, but plateaus below partner self-play on More Fun Coordination and Fun Symmetries Plates (Mixed). These layouts require distinct role responsibilities; alternating roles reveals partner behavior under both, supporting more complete partner-policy estimation and continued improvement. Z2SP w/o Meta+DHR isolates the core online adaptation mechanism from the auxiliary components.

Delayed history replay. Table 3 compares otherwise identical variants in an evaluation batch separate from the main experiment. DHR improves every evaluated post-activation checkpoint, with the largest gain at E_{20} (+40.63) and a positive endpoint gain at E_{100} (+5.88; identity-bootstrap 95% CI $[1.28, 11.50]$). Thus, replay accelerates adaptation, with smaller endpoint gains as the online-only variant catches up.

Table 3: DHR ablation on Counter Circuit. DHR starts in Episode 3, so the variants are identical through E_2. Δ is Z2SP minus Z2SP w/o DHR.

Episode	E_2	E_5	E_{10}	E_{20}	E_{50}	E_{100}
Z2SP w/o DHR	80.10	69.08	86.53	92.83	135.88	156.83
Z2SP	80.10	85.48	101.63	133.45	158.95	162.70
Δ	0.00	+16.40	+15.10	+40.63	+23.08	+5.88

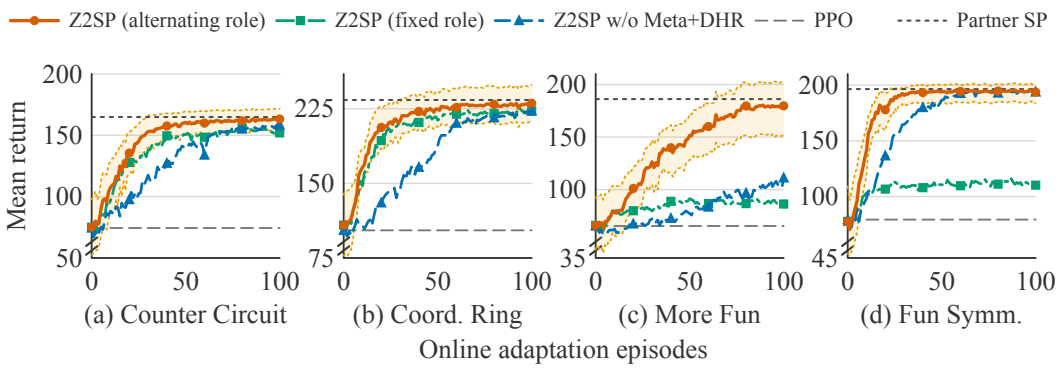


Figure 5: Alternating- versus fixed-role persistent adaptation. Fixed-role results average policies independently adapted in each role; Z2SP w/o Meta+DHR starts from PPO. Shading shows the 95% CI of alternating-role Z2SP.

Meta-initialization over the full adaptation horizon. Figure 6 averages results across all five layouts. During early adaptation, Original Init. drops by 2.24 points at E_1, whereas Meta Init. gains 6.56. By E_5, Meta Init. provides an additional gain of 7.59. It also outperforms Original Init. at all 101 checkpoints and increases normalized AUC from 166.26 to 171.71. The methods approach similar performance at E_{100} (190.63 vs. 192.77), indicating that meta-initialization mainly stabilizes and accelerates early adaptation.

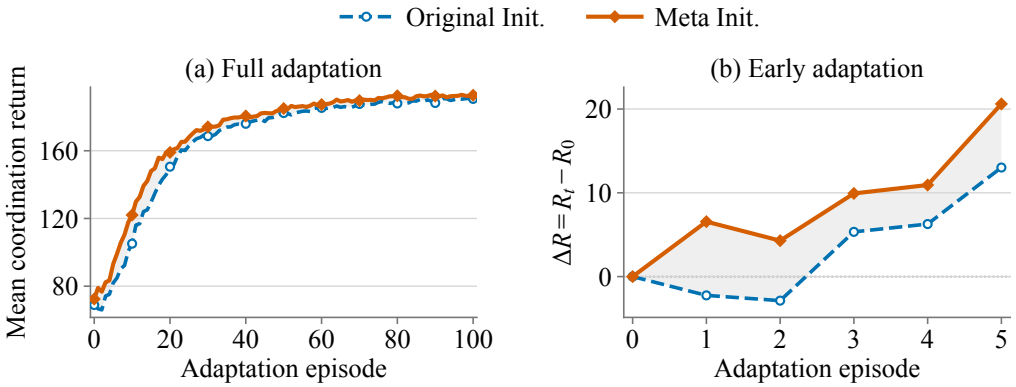


Figure 6: Meta-initialization ablation across five layouts with DHR enabled for both variants: (a) full-horizon return; (b) early return change relative to E_0.

Online computational cost. Z2SP's median amortized cost is 1.31 ms per transition on an RTX A6000 (4.47× PPO-SA), supporting online execution. Measurement details and full results appear in Appendix G.

5 Conclusion

We introduced Z2SP, which improves ZSC by adapting the ego policy toward a finite-sample estimate of the current partner policy from online partner-view observation–action samples. Experiments show that Z2SP progressively narrows the ZSC-to-SP gap, approaches partner self-play, and re-adapts after partner switches.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and language editing, LaTeX formatting, and code development. They were not used to formulate research ideas, design experiments, or analyze experimental results. The authors take responsibility for the final content of this work.

REFERENCES

Aakriti Agrawal, Amrit Singh Bedi, and Dinesh Manocha. Rtaw: An attention inspired reinforcement learning method for multi-robot task allocation in warehouse environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1393–1399. IEEE, 2023.

Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-AI coordination. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Tobias Gessler, Tin Dizdarevic, Ani Calinescu, Benjamin Ellis, Andrei Lupu, and Jakob Foerster. Overcookedv2: Rethinking overcooked for zero-shot coordination. In *International Conference on Learning Representations*, volume 2025, pp. 62752–62784, 2025.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International conference on machine learning*, pp. 4399–4410. PMLR, 2020.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In *International Conference on Machine Learning*, volume 139, pp. 4369–4379. PMLR, 2021.

Bingyu Hui, Lebin Yu, Quanming Yao, Yunpeng Qu, Xudong Zhang, and Jian Wang. Efficient reinforcement learning for zero-shot coordination in evolving games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22110–22118, 2026.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, pp. 2961–2970. PMLR, 2019.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *International Conference on Machine Learning*, volume 267, pp. 27198–27220. PMLR, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

Benjamin Li, Shuyang Shi, Lucia Romero, Huao Li, Yaqi Xie, Woojun Kim, Stefanos Nikolaidis, Charles Lewis, Katia Sycara, and Simon Stepputtis. Adaptively coordinating with novel partners via learned latent strategies. In *Advances in Neural Information Processing Systems*, volume 38, pp. 1268–1295, 2025.

Yang Li, Shao Zhang, Jichen Sun, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Cooperative open-ended learning framework for zero-shot coordination. In *International Conference on Machine Learning*, volume 202, pp. 20470–20484. PMLR, 2023.

Yancheng Liang, Daphne Chen, Abhishek Gupta, Simon S. Du, and Natasha Jaques. Learning to cooperate with humans using generative agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 60061–60087, 2024.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International conference on machine learning*, pp. 7204–7213. PMLR, 2021.

Hadi Nekoei, Xutong Zhao, Janarthanan Rajendran, Miao Liu, and Sarath Chandar. Towards few-shot coordination: Revisiting ad-hoc teamplay challenge in the game of hanabi. In *Conference on Lifelong Learning Agents*, pp. 861–877. PMLR, 2023.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Georgios Papoudakis, Filippos Christianos, and Stefano V. Albrecht. Agent modelling under partial observability for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19210–19222, 2021a.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021b.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10199–10210, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1504–1509, 2010.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14502–14515, 2021.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, volume 119, pp. 9229–9248. PMLR, 2020.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2085–2087. IFAAMAS, 2018.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021a.

Huai-Chih Wang, Hsiang-Chun Chuang, Hsi-Chun Cheng, Dai-Jie Wu, and Shao-Hua Sun. CooT: Learning to coordinate in-context with coordination transformers. In *International Conference on Machine Learning*, volume 306. PMLR, 2026.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent Q-learning. In *International Conference on Learning Representations*, 2021b.

Xue Yan, Jiaxian Guo, Xingzhou Lou, Jun Wang, Haifeng Zhang, and Yali Du. An efficient end-to-end training approach for zero-shot human-ai coordination. In *Advances in Neural Information Processing Systems*, volume 36, pp. 2636–2658, 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624, 2022.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6145–6153, 2023.

This document provides the complete experimental protocols, extended learning curves, disaggregated analyses, and computational measurements supporting the results in the main paper. The organization follows the order of the experiments in the main paper. All test-time adaptation samples are collected online during deployment.

A Experimental Protocol and Reproducibility

A.1 OvercookedV2 Task and Layouts

OvercookedV2 is a cooperative grid-world cooking benchmark in which two chefs share a team return. To complete a dish, they must collect the three ingredients specified by the current recipe, place them in a pot, wait for cooking, collect the dish with a plate, and deliver it to the serving station. At each step, an agent selects one of six discrete actions: four movement directions, stay, or interact. We use full-grid observations and a maximum horizon of 400 steps. This multi-stage pipeline makes performance depend on compatible routing, subtask allocation, object transfer, and action timing. Recipe codes below list the three ingredient types in a dish. Table S1 summarizes how the five layouts emphasize complementary aspects of these capabilities.

Table S1: Structural characteristics of the five evaluation layouts and the coordination capabilities they emphasize. These descriptions distinguish qualitative demands rather than imposing a scalar difficulty ranking.

Layout	Structure	Distinctive coordination demand
Counter Circuit	A shared, elongated circuit around a central counter; either agent can reach the full task pipeline.	Long-range routing, subtask allocation, and compatible circulation, with broader spatial coverage than the smaller ring.
Coordination Ring	A compact shared single-width loop with stations distributed around its perimeter.	Tight movement coordination, blocking and deadlock avoidance, and task sequencing under greater path contention than Counter Circuit.
Forced Coordination	Disconnected work regions: one contains ingredients and plates, while the other contains pots and the serving station.	Mandatory object handoffs and complementary roles; it most directly enforces structural interdependence.
More Fun Coordination	Disconnected rooms with complementary ingredient sources, two shared pots, and recipes 011 or 022.	Recipe-conditioned complementary roles, cross-room synchronization, and parallel cooking; unlike Forced Coordination, the key asymmetry lies in indispensable recipe ingredients rather than exclusive processing stations.
Fun Symmetries Plates (Mixed)	Plates and the recipe indicator lie on one side, ingredients and delivery on the other, with a single shared pot and recipes 001 or 011.	Ordered handoffs through a single cooking bottleneck and convention alignment under spatial and recipe symmetry; role coverage matters more than in the shared-circuit layouts.

A.2 Evaluation Partners

The evaluation set contains 20 self-play-trained partner policies, with five policies from each of IQL, VDN, MAAC, and HAPPO. For presentation, we index the five policies within each family as IQL1–5, VDN1–5, MAAC1–5, and HAPPO1–5; these display aliases are used consistently throughout the supplementary figures. These policies are disjoint from the MAPPO, WQMIX, and QPLEX population used for meta-initialization.

Each comparison method is trained with five independent runs using five random seeds derived from the root seed 42. The main five-layout persistent-adaptation curves average five independent adap-

tation runs with seeds 1701–1705 for every partner. More Fun Coordination is recipe dependent; its reported values give equal weight to recipes 011 and 022 at every evaluation checkpoint.

A.3 Persistent-Adaptation Protocol

For the main persistent-adaptation experiment, Z2SP interacts with the same partner for 100 episodes. The environment state is reset at every episode boundary, whereas the ego actor parameters, Adam state, and replay buffer are retained. The ego role alternates between agent 0 and agent 1 across episodes and remains fixed within an episode. Every saved policy is evaluated independently in both deployment roles using 20 fixed evaluation keys per role; these evaluations do not modify the adaptation state.

We use E_m to denote evaluation after m completed adaptation episodes. Thus, E_0 is evaluated before receiving samples from the current partner, and E_1 is evaluated after one complete adaptation episode. "Final 20" denotes the arithmetic mean of $E_{81}, \ldots, E_{100}$.

The baseline comparison reports Z2SP before adaptation and after one adaptation episode. Z2SP at E_1 is evaluated in separate rollouts without policy updates; it is not the return collected during the adaptation episode.

A.4 Implementation Details

Z2SP starts from the official CNN-based PPO actor in OvercookedV2 and updates only the actor while keeping the critic frozen. Comparison methods are trained for 10^7 environment steps, and episodes last 400 steps. Test-time Adam uses learning rate 1.7919×10^{-5} and global clipping at 10. DHR starts after Episode 2 with $B = 256$, $m_o = 4$, $m_r = 3$, and $\beta = 0.5$. Meta-initialization uses two closed-loop rounds of 300 outer updates with $\lambda_{\mathrm{ret}} = 1$, $\lambda_{\mathrm{init}} = 10$, and $\lambda_{\mathrm{anc}} = 10^{-4}$.

A.5 Aggregation and Statistical Estimation

For the five-layout experiment, returns are first averaged over the evaluation runs, the two deployment roles, and the five adaptation runs for each partner identity. The reported layout mean then averages the 20 partner identities. Unless otherwise specified, the corresponding 95% interval is

$$\bar{R} \pm 1.96\, s_R/\sqrt{N}, \qquad N = 20, \tag{S1}$$

where s_R is the sample standard deviation across partner identities. The overall result is the unweighted mean of the five layout means.

In addition to the main paper's SP-normalized return $S_{\ell,m} = 100 R_{\ell,m}/R_{\ell,\mathrm{SP}}$, supplementary analyses report gap recovery after m episodes on layout ℓ:

$$G_{\ell,m} = 100 \frac{R_{\ell,m} - R_{\ell,0}}{R_{\ell,\mathrm{SP}} - R_{\ell,0}}. \tag{S2}$$

Here, $R_{\ell,0}$ is the initial ZSC return. SP-normalized return measures the attained fraction of partner SP performance, whereas gap recovery measures the fraction of the initial ZSC-to-SP gap that has been recovered.

B Extended Five-Layout Results

Figure S1 reports the complete raw-return trajectories corresponding to the SP-normalized return curves in the main paper. The five layouts exhibit different adaptation rates, but all approach their partner self-play references over persistent adaptation. Counter Circuit, Coordination Ring, Forced Coordination, and Fun Symmetries Plates recover rapidly, whereas More Fun Coordination requires a longer adaptation horizon.

By E_{40}, four layouts have already captured most of their eventual improvement. More Fun Coordination continues to improve substantially, indicating slower adaptation rather than a lower final coordination level.

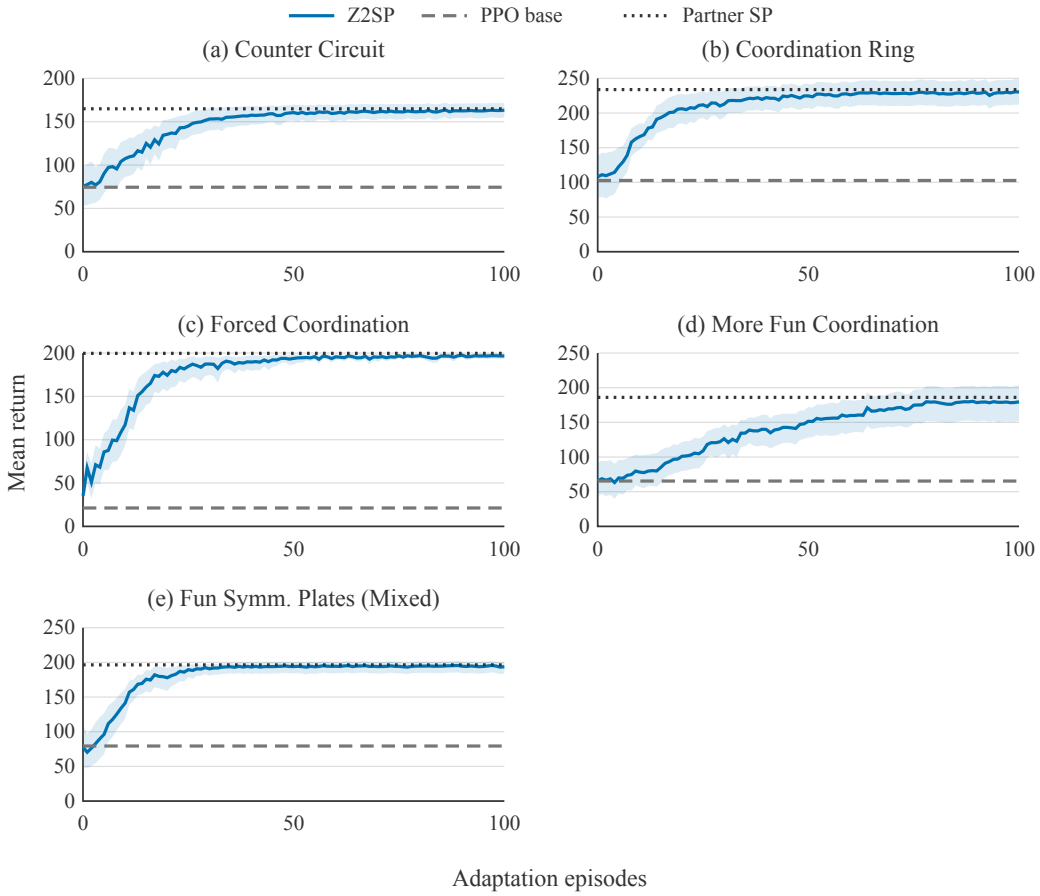


Figure S1: Raw team return from E_0 through E_{100} on the five OvercookedV2 layouts. Horizontal references show the PPO base and partner self-play return; shading denotes 95% confidence intervals. Curves average five adaptation runs, the 20 evaluation partners, and both deployment roles.

C Partner-Level Persistent-Adaptation Results

Layout averages can conceal substantial variation among partners. Figure S2 therefore disaggregates the main alternating-role experiment at four measured checkpoints, without interpolation. The heatmaps expose layout- and partner-specific differences hidden by aggregate curves.

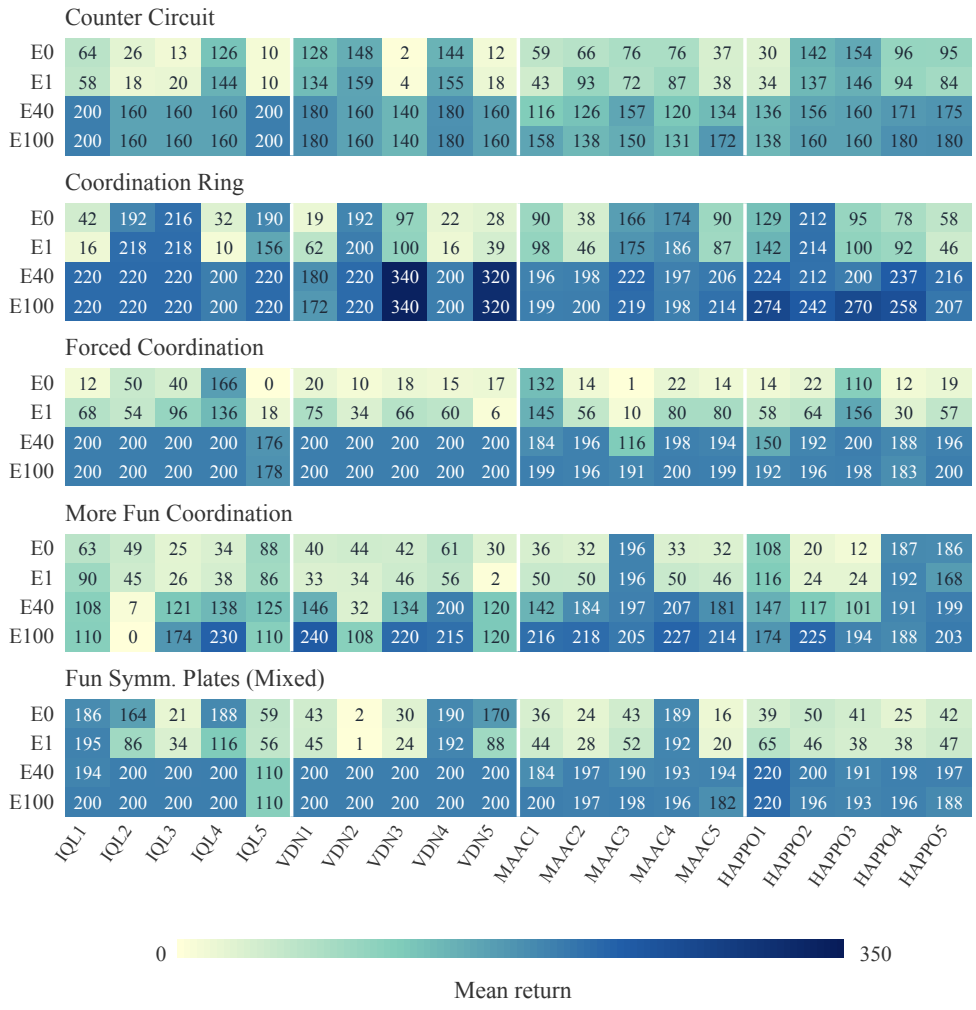


Figure S2: Partner-level returns for Z2SP in the alternating-role main experiment. Rows are measured checkpoints and columns are the 20 partners, grouped by family; values average both deployment roles.

Figures S3–S7 report all 101 measured checkpoints for Z2SP w/o Meta+DHR and Z2SP under the same alternating-role protocol. The ablated variant retains only online partner-view adaptation from PPO. The curves reveal transient regressions and heterogeneous adaptation rates hidden by layout averages; every point is measured rather than interpolated.

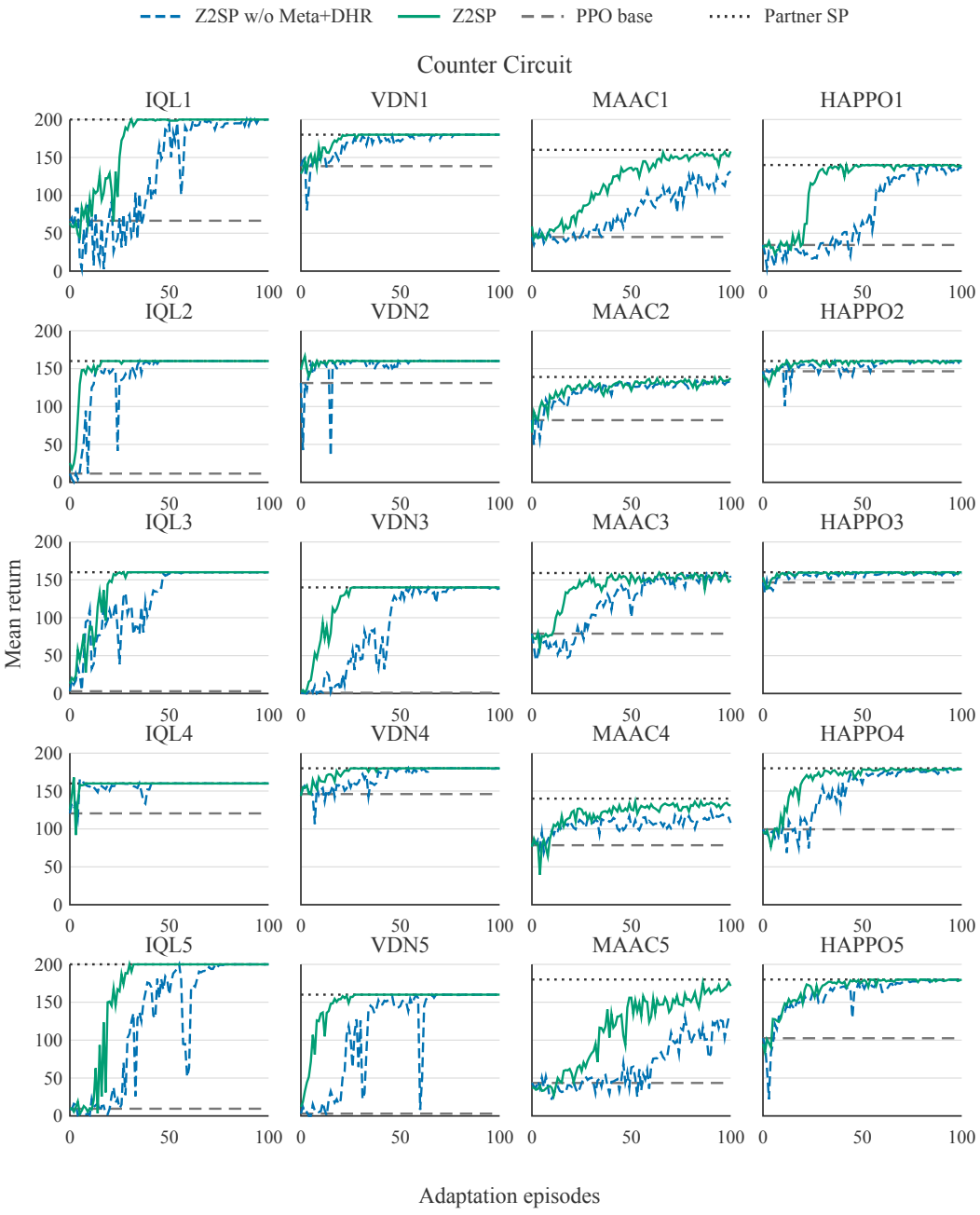


Figure S3: Dense partner-level Counter Circuit curves from the alternating-role main experiment, comparing Z2SP w/o Meta+DHR with Z2SP. Every point is measured and averages both deployment roles.

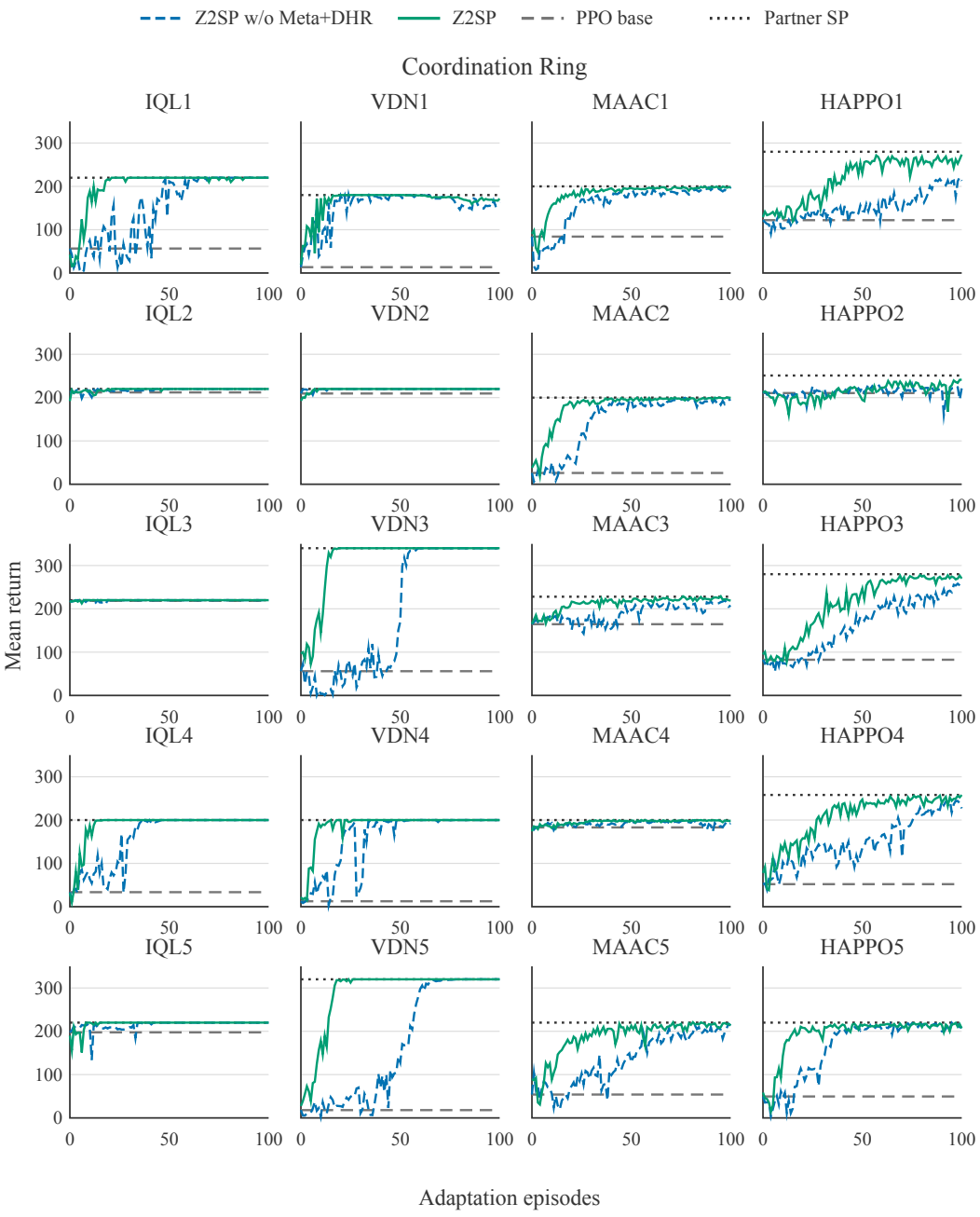


Figure S4: Dense partner-level Coordination Ring curves from the alternating-role main experiment: Z2SP w/o Meta+DHR versus Z2SP, under the protocol of Figure S3.

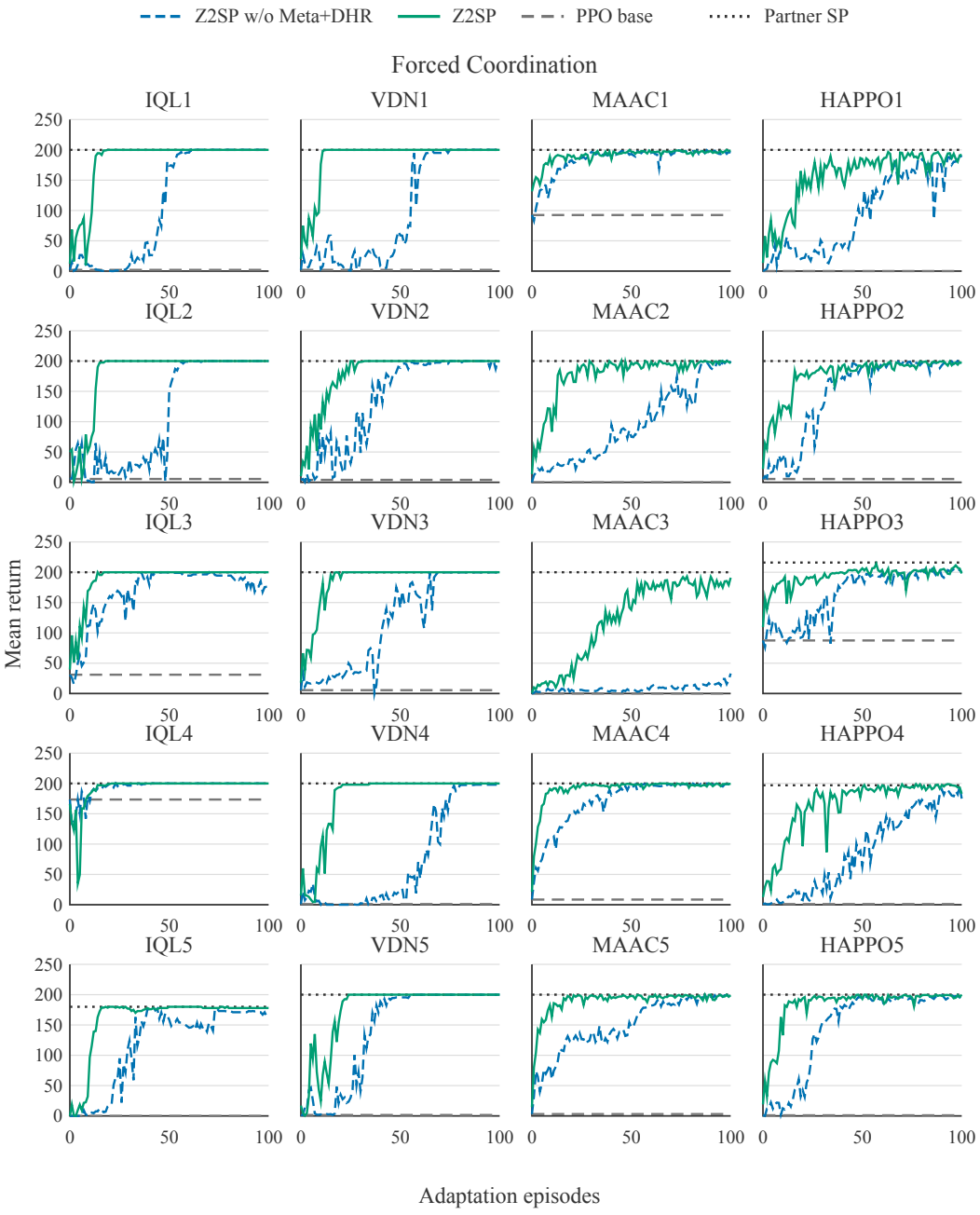


Figure S5: Dense partner-level Forced Coordination curves from the alternating-role main experiment: Z2SP w/o Meta+DHR versus Z2SP, under the protocol of Figure S3.

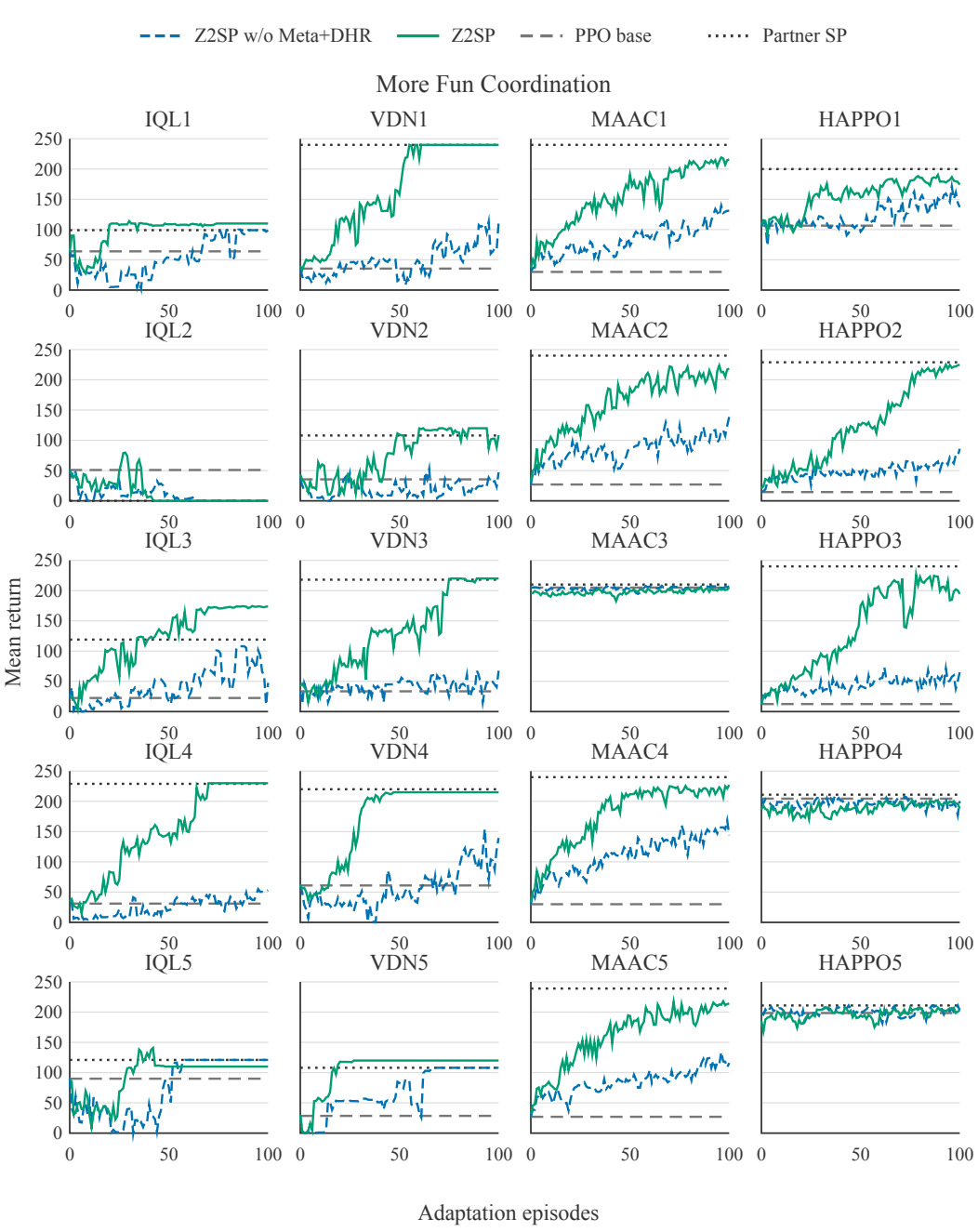


Figure S6: Dense partner-level More Fun Coordination curves from the alternating-role main experiment: Z2SP w/o Meta+DHR versus Z2SP, under the protocol of Figure S3.

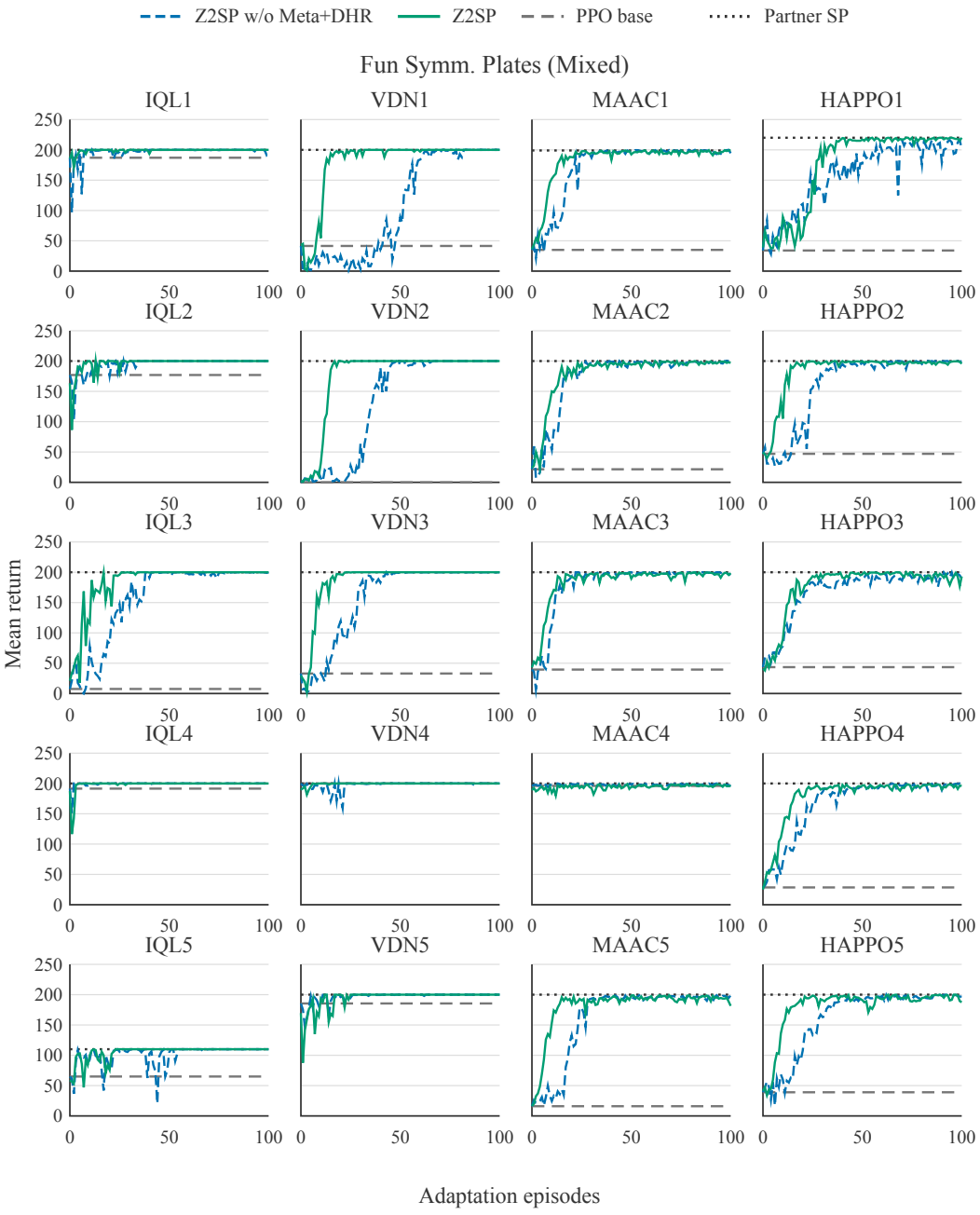


Figure S7: Dense partner-level Fun Symmetries Plates (Mixed) curves from the alternating-role main experiment: Z2SP w/o Meta+DHR versus Z2SP, under the protocol of Figure S3.

D CONTINUAL RE-ADAPTATION UNDER PARTNER SWITCHES

The experiment in the main paper uses a controlled $A \to B \to A$ sequence on Counter Circuit. A single Z2SP policy interacts with each stage partner for 50 episodes. The actor, Adam state, and replay buffer are retained at both switches. Roles alternate across episodes and remain fixed within each episode. We evaluate four family-balanced partner-switch sequences and three adaptation runs, producing 12 sequence–run jobs under the same checkpoint evaluation protocol.

Figure S8 disaggregates the mean curve by partner sequence. Although the return falls immediately after each switch, every sequence subsequently recovers toward the self-play level of the new partner. This confirms that specialization is partner specific but remains reversible through continued online adaptation.

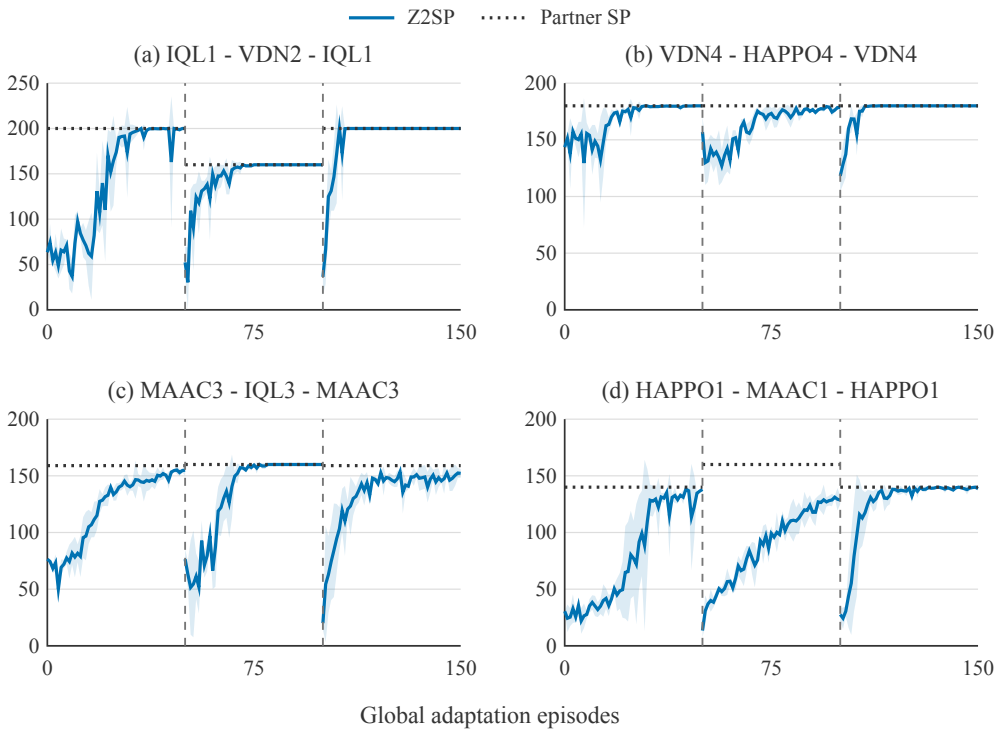


Figure S8: Disaggregated $A \to B \to A$ re-adaptation curves on Counter Circuit. Each panel averages three adaptation seeds and both deployment roles; shading denotes 95% confidence intervals across adaptation seeds. Vertical lines mark switches, and dotted horizontal segments show the active partner's self-play return.

For a stage starting at global checkpoint s, we define the local recovery at local episode u by replacing $R_{\ell,0}$ in Eq. (S2) with the return at the beginning of that stage:

$$G_u^{\text{stage}} = 100 \frac{R_{s+u} - R_s}{R_{\text{SP}} - R_s}. \tag{S3}$$

Table S2 reports cluster-bootstrap intervals over the 12 sequence–seed jobs. Revisiting A reaches 90% stage-local gap recovery faster than the initial visit. We interpret this as faster reacquisition under this protocol rather than as a standalone causal proof of long-term memory.

E ROLE COVERAGE DURING PERSISTENT ADAPTATION

The main protocol alternates the ego role across episodes. To isolate the effect of role coverage, we also adapt two independent fixed-role policies: one always acts as agent 0 and one always acts as

Table S2: Stage-local gap recovery. The respective return gains and 95% cluster-bootstrap intervals are $89.75\,[68.75, 110.25]$, $81.96\,[59.29, 102.13]$, and $117.17\,[95.08, 138.33]$.

Stage	Initial	Final	Partner SP	Recovery	Episodes to $G^{\text{stage}} = 90\%$
A (initial)	78.50	168.25	169.75	98.36%	31
B	74.79	156.75	165.00	90.85%	43
A (revisit)	50.54	167.71	169.75	98.29%	11

agent 1. Each fixed-role policy is evaluated in both deployment roles. Their reported mean is

$$\frac{1}{2}\left[\frac{R_{0\to0} + R_{0\to1}}{2} + \frac{R_{1\to0} + R_{1\to1}}{2}\right], \tag{S4}$$

where $R_{i\to j}$ evaluates a policy adapted in role i while deployed in role j. We additionally report the strict same-role mean $\frac{1}{2}(R_{0\to0} + R_{1\to1})$.

Table S3: Fixed-role results under two evaluation conventions. Final 20 averages E_{81}–E_{100}.

Layout	Both-role evaluation	Same-role evaluation
	E_{40}/E_{100}/Final 20	E_{40}/E_{100}/Final 20
Counter	$149.58/151.95/153.99$	$144.53/143.08/146.61$
Coord. Ring	$211.23/223.03/221.02$	$205.73/217.45/213.46$
More Fun	$88.86/86.33/88.23$	$49.13/44.50/43.83$
Fun Symm.	$108.04/110.28/112.32$	$68.40/55.75/57.56$

Table S3 shows that evaluating a fixed-role policy only in its adaptation role can substantially underestimate how much partner information it has acquired for the other deployment role. Nevertheless, More Fun Coordination and Fun Symmetries Plates remain much harder under fixed-role adaptation. Their stronger role specialization makes samples from both responsibilities important for constructing a more complete approximation of the partner policy.

F Extended Component Ablations

F.1 Delayed History Replay

The DHR ablation holds all other components and evaluation settings fixed. Z2SP without DHR performs one Adam update from the latest partner-view sample at each valid transition. Z2SP performs the same update and, beginning in Episode 3, an additional replay update using the current sample, three recent samples, and four approximately uniformly spaced older samples.

Table S4 shows positive paired gains after replay activates, with the largest benefit during mid-adaptation and a positive endpoint advantage under both bootstrap schemes.

Table S4: DHR ablation on Counter Circuit (Z2SP minus no-DHR).

E_m	Difference	Identity 95% CI	Identity+seed 95% CI
E_5	+16.40	$[3.13, 31.88]$	$[2.55, 31.88]$
E_{10}	+15.10	$[3.03, 30.18]$	$[2.27, 30.85]$
E_{20}	+40.63	$[21.33, 62.75]$	$[20.80, 63.25]$
E_{50}	+23.08	$[12.33, 35.60]$	$[11.82, 36.28]$
E_{100}	+5.88	$[1.28, 11.50]$	$[1.10, 11.98]$

F.2 Meta-Initialization

We extend the main alternating-role meta-initialization ablation across all five layouts. Both conditions use the same online update and DHR and differ only in their initialization.

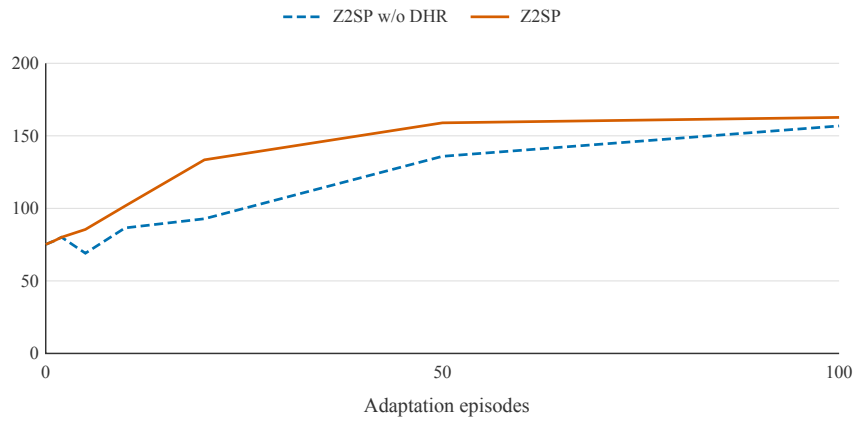


Figure S9: Counter Circuit returns with and without DHR. The variants coincide through E_2 because replay starts after Episode 2.

Meta-initialization remains above the original initialization at all 101 macro-averaged checkpoints. At E_1, the original initialization falls 2.24 points from E_0, whereas meta-initialization gains 6.56; by E_5, their respective gains are 13.02 and 20.61. As reported in Table S5, meta-initialization increases the normalized E_0–E_{100} AUC on every layout and raises the five-layout average by 5.45 points.

Table S5: Normalized E_0–E_{100} AUC under the main alternating-role protocol.

Layout	Original Init.	Meta Init.	Difference
Counter	145.59	147.60	+2.01
Coord. Ring	205.80	211.30	+5.50
Forced	170.63	178.03	+7.40
Fun Symm.	181.27	181.87	+0.60
More Fun	128.02	139.76	+11.74
Overall	166.26	171.71	+5.45

G Detailed Online Computational Cost

We measure steady-state amortized throughput on one NVIDIA RTX A6000 using Counter Circuit and IQL2. Each condition runs three 100-episode trials with 399 valid transitions per episode. Timing includes environment interaction, policy execution, and online updates, synchronizes the GPU after each episode, and excludes one-time JIT compilation. Meta-initialization is performed offline and adds no online computation.

Table S6: Steady-state runtime over 39,900 valid transitions.

Method	Median time	ms/step	Steps/s	× PPO-SA
PPO-SA base	11.68 s	0.293	3416	1.00
Z2SP w/o DHR	27.70 s	0.694	1440	2.37
Z2SP	52.26 s	1.310	763	4.47

Per-run times are $[11.68, 11.63, 11.84]$ seconds for PPO-SA, $[34.51, 27.47, 27.70]$ for Z2SP without DHR, and $[63.60, 52.00, 52.26]$ for Z2SP. Each Z2SP run executes 39,102 replay updates; its median cost is 1.31 ms per valid transition. These results measure amortized simulator throughput rather than worst-case physical-control latency.

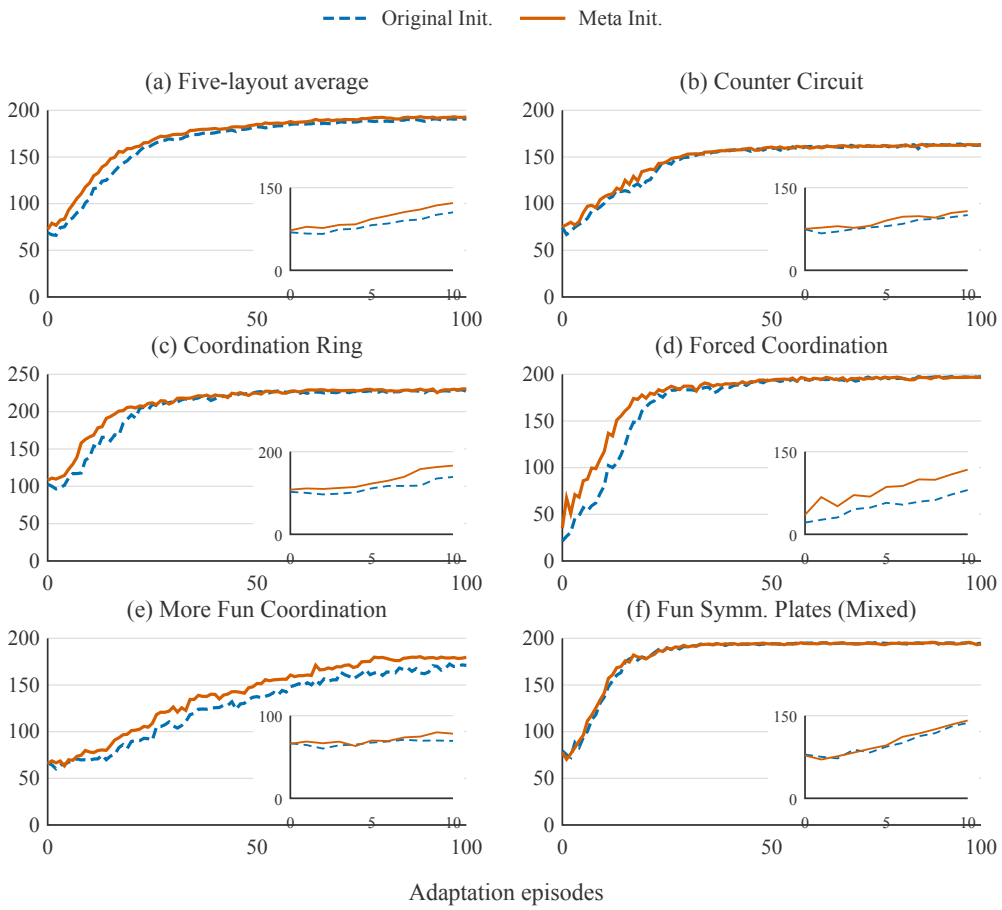


Figure S10: Five-layout alternating-role meta-initialization ablation with DHR enabled for both variants. Panel (a) gives the macro average; the remaining panels show individual layouts, with insets enlarging E_0–E_{10}.

H Scope and Limitations

Z2SP is currently evaluated in two-agent, fully observable OvercookedV2 tasks with compatible observation encodings and a common discrete action space, but its update is not intrinsically restricted to two agents. In an N-agent team whose other agents share a policy, the ego can pool all available teammate samples using

$$\mathcal{L}_t^{\text{in}}(\theta_t) = -\frac{1}{N-1}\sum_{i\neq e}\log \pi_{\theta_t}^{e}(a_t^i \mid o_t^i), \tag{S5}$$

where e indexes the ego agent, thereby estimating their common policy through the same online update. If teammates instead follow heterogeneous policies or occupy nonexchangeable roles, naively pooling samples would conflate different targets; an extension would require teammate- or role-conditioned policies or separate estimates.

Z2SP uses the partner observation–action histories available in the fully observable tasks considered here. In partially observable settings, if partner observations or executed actions are unavailable, additional inference or alternative supervision would be required. The finite-sample objective approximates the partner policy only over observations covered during interaction and therefore does not guarantee monotonic return improvement. Moreover, moving ZSC toward self-play is most beneficial when the partner's self-play return exceeds its initial ZSC return. Empirical evaluation in larger teams, partially observable settings, heterogeneous or continuous action spaces, and human–AI teams remains future work.