COOT .: LEARNING TO COORDINATE IN-CONTEXT WITH COORDINATION TRANSFORMERS

Anonymous authors

000

001

003

006

008 009

010 011

012

013

014

016

017

018

019

021

024

025

026

027 028

029

031

033

034

037

040

041

042

043

044

046

047

048

050 051

052

Paper under double-blind review

ABSTRACT

Effective coordination among artificial agents in dynamic and uncertain environments remains a significant challenge in multi-agent systems. Existing approaches, such as self-play and population-based methods, either generalize poorly to unseen partners or require impractically extensive fine-tuning. To overcome these limitations, we propose Coordination Transformers (CooT), a novel in-context coordination framework that uses recent interaction histories to rapidly adapt to unseen partners. Unlike prior approaches that primarily aim to diversify training partners, CooT explicitly focuses on adapting to new partner behaviors by predicting actions aligned with observed interactions. Trained on trajectories collected from diverse pairs of agents with complementary preferences, CooT quickly learns effective coordination strategies without explicit supervision or parameter updates. Across diverse coordination tasks in Overcooked, CooT consistently outperforms baselines including population-based approaches, gradient-based fine-tuning, and a Meta-RL-inspired contextual adaptation method. Notably, fine-tuning proves unstable and ineffective, while Meta-RL struggles to achieve reliable coordination. By contrast, CooT achieves stable, rapid in-context adaptation and is consistently ranked the most effective collaborator in human evaluations.

1 Introduction

Coordination is fundamental to intelligent behavior, enabling individuals to achieve shared goals through joint effort. In dynamic and uncertain environments, such as team sports or traffic navigation, humans adjust their actions based on others' behaviors and intentions, facilitating effective collaboration. Replicating this adaptive coordination in artificial agents remains a core challenge in multi-agent systems (Cao et al., 2012; Zhang et al., 2021), especially in domains like robotics (Yan et al., 2013), gaming (Matignon et al., 2012), and human-AI interaction (Carroll et al., 2019), where an agent's success often depends on effectively responding to its partners.

Existing methods for developing agents capable of coordinating with unseen partners have pursued various strategies. Self-play (Tesauro, 1994; Yu et al., 2022a; Hu et al., 2021; Wang et al., 2023) trains agents by having them repeatedly interact with copies of themselves. Although this is effective in coordinating with known partners, it frequently leads to conventions that fail when interacting with unfamiliar collaborators. Population-based methods (Jaderberg et al., 2017) seek to address this by training agents within diverse populations using strategies such as partner randomization (Hu et al., 2020; Lucas & Allen, 2022), reward shaping (Yu et al., 2023), explicit modeling of partner behaviors (Lou et al., 2023), and enhancing behavioral diversity (Zhao et al., 2023; Lupu et al., 2021; Xue et al., 2024). However, while these approaches promote robustness, they often lack mechanisms for efficient online adaptation, which limits their applicability in open-ended or real-world scenarios. Attempting to fine-tune multi-agent reinforcement learning (MARL) policies on new partners also proves ineffective, since adaptation often demands thousands or even millions of interaction trajectories to learn effective coordination strategies (Nekoei et al., 2023), thus making few-shot adaptation infeasible.

Recent advances in in-context learning (ICL; Brown et al., 2020; Wei et al., 2022; Li et al., 2023b) offer a promising alternative by enabling models to condition their behavior on contextual examples, without additional training or fine-tuning. Originally demonstrated in language modeling (Brown et al., 2020), ICL has recently been adapted to sequential decision-making domains, including offline

reinforcement learning (Chen et al., 2021; Lee et al., 2023) and distillation of reinforcement learning policies (Laskin et al., 2023; Kirsch et al., 2023). However, applying ICL to coordination introduces distinct challenges. Unlike task-focused settings with well-defined rewards, coordination requires aligning with diverse partner behaviors, often without explicit feedback or well-defined success metrics. Thus, the primary challenge shifts from generalizing across tasks to effectively interpreting and adapting to the behaviors of new partners purely based on observed interactions.

To address this challenge, we introduce Coordination Transformers (COOT), a framework designed specifically for in-context partner adaptation. Whereas prior ICL-inspired methods (Chen et al., 2021; Laskin et al., 2023) primarily target task generalization, COOT emphasizes generalization across diverse partner behaviors, a challenge less directly addressed in existing work. Specifically, COOT predicts actions that best align with the observed partner's behavior to maximize collaboration effectiveness. To achieve this, we train COOT on trajectories collected from interactions between pairs of agents whose behaviors reflect distinct underlying preferences. Such agents, which we term behavior-preferring agents, operate based on event-based hidden reward functions that guide their actions but are hidden from their partners. These event-based preferences yield diverse and potentially unpredictable coordination behaviors, even within identical environments. By observing interactions between these agents and their complementary counterparts, COOT learns context-driven strategies that enable effective coordination with previously unseen partners.

We evaluated CooT across a broad set of coordination methods, including population-based approaches (Yu et al., 2023; Zhao et al., 2023), gradient-based fine-tuning, and a Meta-RL—inspired baseline (Rakelly et al., 2019) with contextual adaptation, using the Overcooked environment (Carroll et al., 2019), a popular multi-agent benchmark requiring precise and adaptive coordination. Results demonstrate that CooT consistently outperforms these baselines in diverse coordination tasks involving unseen agents. Notably, gradient-based fine-tuning proved unstable and delivered limited improvements, while the contextual adaptation Meta-RL baseline performed poorly, failing to achieve reliable coordination. By contrast, CooT achieves stable, rapid adaptation in context without parameter updates. Furthermore, human evaluations consistently rank CooT as the most preferred partner in interactive collaboration. Ablation studies provide additional insights, revealing that CooT improves coordination by observing more interactions with novel partners and that preserving temporal structure in interaction contexts is critical for generalization. Overall, these results demonstrate CooT 's effectiveness for adapting to diverse, unseen partners in cooperative settings.

2 RELATED WORK

Learning to coordinate. Learning to coordinate in multi-agent reinforcement learning (MARL) involves designing agents that can develop effective collaboration strategies, often under varying assumptions about their partners. One key challenge in coordination is zero-shot coordination (ZSC) (Stone et al., 2010). Without further training, ZSC requires an ego agent to cooperate with previously unseen partners at test time. Self-play (SP) (Tesauro, 1994; Yu et al., 2022a; Wang et al., 2023) trains agents by interacting with copies of themselves, but they often struggle to generalize to novel partners (Carroll et al., 2019), due to overfitting to a fixed partner policy. To improve robustness, population-based methods (Jaderberg et al., 2017; Long* et al., 2020; Hu et al., 2020; Strouse et al., 2021) expose the ego agent to a range of partner behaviors during training. Several studies continue increasing population diversity using entropy-based methods (Zhao et al., 2023; Lupu et al., 2021), reward shaping (Lucas & Allen, 2022), off-belief learning (Hu et al., 2021), groups of population (Xue et al., 2024), hidden-utility functions to simulate biased or human-like preferences (Yu et al., 2023), and training with incompatible agents (Charakorn et al., 2023; Li et al., 2023a). Despite these advances, most coordination strategies still require extensive training interactions. To address this, few-shot coordination (Nekoei et al., 2023) has emerged as a promising direction, aiming to enable agents to adapt to new partners with minimal experience rapidly. Approaches like PECAN (Lou et al., 2023) allow agents to analyze and respond to observed partner behavior for better coordination. Building on this idea, our proposed method, COOT, leverages the in-context learning capabilities of transformers to achieve effective few-shot coordination, offering a scalable solution for generalizing coordination strategies to novel partners.

In-context Reinforcement Learning. In-context learning (ICL) (Brown et al., 2020; Wei et al., 2022; Li et al., 2023b) enables models to adapt at inference by conditioning on examples rather than gradient

updates. Beyond its success in language and vision (Brown et al., 2020; Yu et al., 2022b), recent studies extend ICL to reinforcement learning (Chen et al., 2021; Jing et al., 2023; Laskin et al., 2023; Lee et al., 2023), showing applications in offline RL, online decision-making, opponent modeling, and meta-RL. Prior works suggest that transformers may implicitly implement optimization-like procedures (Von Oswald et al., 2023), with trajectories serving as contextual examples for learning algorithms or posterior sampling (Laskin et al., 2023; Lee et al., 2023; Jing et al., 2023). Unlike these task-focused approaches, CooT targets coordination, generalizing to unseen partners by interpreting prior interaction trajectories as context.

Transfer Learning in Reinforcement Learning. Transfer learning in reinforcement learning (RL) (Duan et al., 2016; Gupta et al., 2017; Finn et al., 2017; Tobin et al., 2017; Zhang et al., 2020; Xing et al., 2021) seeks to reuse knowledge from prior tasks to improve performance on new ones with minimal training, a key goal when task-specific experience is costly. Approaches include task-invariant representations (Gupta et al., 2017; Xing et al., 2021), meta-learning for rapid adaptation (Duan et al., 2016; Finn et al., 2017; Rakelly et al., 2019), and domain randomization (Tobin et al., 2017). Recent work highlights conceptual links to ICL (Laskin et al., 2023; Lee et al., 2023): both aim to generalize across conditions, but many transfer RL works rely on explicit parameter updates, while ICL adapts a fixed model by conditioning on interaction histories. This implicit form of transfer enables flexible, sample-efficient coordination. Building on this, CooT learns offline representations and adapts to novel online partners within only a few episodes.

3 Preliminary

3.1 HIDDEN-UTILITY MARKOV GAME

We formulate the in-context coordination problem as a Hidden-Utility Markov Game (HU-MG), inspired by the framework introduced in Hidden-Utility Self-Play (Yu et al., 2023). The HU-MG builds on the two-agent decentralized Markov game (Bernstein et al., 2002), defined as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}_t)$, where \mathcal{S} denotes the state space, $\mathcal{A} = \mathcal{A}^i \times \mathcal{A}^w$ is the joint action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A}$ represents the transition function, and \mathcal{R}_t the shared global reward. Both agents, π_i and π_w , aim to maximize \mathcal{R}_t .

A two-agent HU-MG is instead defined as $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}_t, \mathcal{R}_w)$, where \mathcal{R}_w is a hidden reward space containing functions $r^w \sim \mathcal{R}_w$. The hidden reward r^w , observable only by π_w , is formulated as linear functions over event features, inducing distinctive behavioral preferences or conventions. Note that π_w corresponds to the partner policy π_i^p introduced in Section 4, but we adopt the new notation to better distinguish partner indices in the dataset setting.

3.2 In-Context Learning with Decision-Pretrained Transformers

In-context learning refers to a model's ability to generalize by conditioning predictions on contextual examples at inference time. A Decision-Pretrained Transformer (DPT) (Lee et al., 2023) trains on sequences of (s, a, r) triplets, representing states, actions, and returns. At inference, it autoregressively predicts the optimal action for a given state, conditioned on contextual examples. This reframes reinforcement learning as sequence modeling, where task identity is inferred from context. However, standard DPT assumes access to extrinsic rewards and explicit task objectives, limiting applicability in coordination settings with diverse and implicit partner preferences.

4 METHOD

To build agents capable of efficient and adaptive coordination with unseen partners, we propose Coordination Transformer (CooT ②), a framework that adapts to novel partners by conditioning on past interactions. Our approach involves training a transformer on trajectories collected from diverse partners and their corresponding best responses, i.e., the behavior that yields the highest expected return given a specific partner's behavior, enabling effective generalization to previously unseen collaborators. We begin by formalizing coordination as a Hidden-Utility Markov Game (HU-MG) in Section 4.1, where agents behave according to hidden, agent-specific reward functions. We then describe the process of generating a diverse dataset of interaction histories in Section 4.2, explain

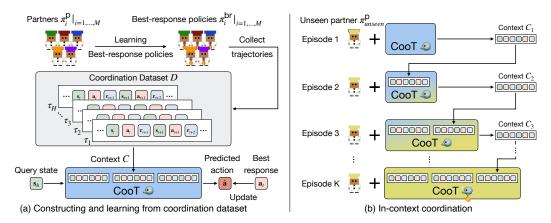


Figure 1: **COOT.** (a) **Training.** We generate a dataset \mathcal{D} of trajectories between behavior-preferring agents and their best-response (BR) policies. For each training instance, COOT receives query states \mathbf{s}_h and context \mathbf{C} of past interactions, and learns to predict an action \mathbf{a} mimicking the BR action $\hat{\mathbf{a}}$. (b) **Evaluation.** At test time, COOT coordinates with unseen partners by continually updating its context from recent episodes, which adapts to the partner online without gradient updates, enabling few-shot generalization through context update alone.

how CooT is trained to predict partner-aligned actions in Section 4.3, and present our deployment protocol for evaluating generalization in Section 4.4. An overview of CooT is illustrated in Figure 1.

4.1 From Task Generalization to Coordination Generalization

Recent advances in in-context reinforcement learning enable sequence models to generalize across tasks by conditioning on prior experiences. These methods assume explicit task definitions and observable rewards, which allow models to infer goals or dynamics from context. However, they do not address the challenge of adapting to diverse partners with hidden preferences under the same task.

We reframe this challenge as a two-player Hidden-Utility Markov Game (HU-MG), where one of the agent's behaviors are driven by a hidden reward function that is unobservable to its partner. We refer to such agents as *behavior-preferring agents*, since their preferences are revealed only through behavior. These hidden utilities induce diverse coordination patterns, even when the environment remains fixed. In this setting, adapting to a partner requires inferring its underlying preferences through observed behavior.

Our approach treats partner adaptation as an in-context learning problem. Rather than relying on explicit task labels or extrinsic rewards, the model conditions on past interactions to infer hidden utilities. In contrast to ICRL, where the goal is to recover the optimal policy for each task, in HU-MG, the objective becomes learning a *best-response policy* tailored to the inferred partner behavior.

4.2 Dataset Generation

To enable in-context adaptation to novel partners, we construct a dataset D containing interaction histories between behavior-preferring agents and their corresponding best responses. Our data generation process is grounded in the HU-MG framework as described in Section 3.1, where coordination challenges arise from agent-specific hidden reward differences.

Following ZSC-Eval (Wang et al., 2024), we simulate hidden-utility preferences by first defining a set of discrete environmental events. We then construct different hidden reward functions using linear combinations of the environmental events to form a space of reward \mathcal{R}_w . For each hidden reward function $r_i^w \sim \mathcal{R}_w$, we jointly train a behavior-preferring partner policy π_i^p , which is driven by r_i^w , and its best-response policy π_i^{br} using Proximal Policy Optimization (PPO) (Schulman et al., 2017).

To ensure diversity among training partners, we compute an event-based diversity score d_i for each best-response policy $\pi_i^{\rm br}$, and then select the top-N diverse pairs to form the training pool $\Pi_{\rm train}$. For

each selected pair (π_i^p, π_i^{br}) , we collect T trajectories to form a fixed-length context C. From each context, we sample query states s_h and construct tuples of the form (s_h, C, \hat{a}) , where $\hat{a} = \pi_i^{br}(s_h)$ is the best-response action. These tuples are added to the dataset D for training CooT.

To construct the evaluation set, we extract behavioral features ϕ_i from trajectories generated by each $\pi_i^{\rm br}$ and compute the population diversity. To efficiently select a diverse evaluation set, we then apply Determinantal Point Process sampling (Kulesza et al., 2012) over \mathbf{K} , a similarity matrix of behavioral feature set $\{\phi_i\}$, resulting in a behaviorally diverse evaluation set $\Pi_{\rm eval} \subset \Pi_0$.

4.3 Training

Given the dataset of context–query–action tuples, we train the Coordination Transformer (CooT) to perform best-response prediction via in-context learning. At each training step, the model receives query states s_h and a context window C, consisting of past interactions between a behavior-preferring agent and its best response. Conditioned on this context, the transformer predicts the best-response action distribution $\hat{p}_h(\cdot) = M_\theta(\cdot \mid s_h, C)$.

The model is trained to minimize the cross-entropy loss: $\mathcal{L} = -\log \hat{p}_h(\hat{a})$ where $\hat{a} = \pi_i^{\rm br}(s_h)$ is the ground-truth best-response action given the query states. This objective encourages the model to map partner behaviors to effective responses, supporting generalization through in-context adaptation to unseen collaborators. Further training details are provided in Appendix B.2

4.4 Online Deployment

At test time, CooT is deployed with unseen behavior-preferring partner policies $\pi_{unseen}^p \sim \Pi_{\text{eval}}$ over multiple episodes of length Z. A fixed-length context C, composed of T previously collected trajectories $\tau = (s_t, a_t, r_t)_{t=1}^Z$, is initialized before deployment begins.

At each timestep t in an episode of length Z, the agent observes the context C, current state s_t , and a short history of prior states (e.g., $\{s_{t-n},\ldots,s_t\}$), which together form the query state s_h . Based on this input, CooT predicts an action distribution $\hat{p}_t(\cdot) = M_{\theta}(\cdot \mid s_h, C)$ and samples an action $a_t \sim \hat{p}_t$.

After executing action a_t and observing reward r_t , the transition (s_t, a_t, r_t) is recorded. Once the episode is complete, the resulting trajectory of Z steps is appended to the context buffer. Repeating this process across E episodes allows CooT to refine its coordination strategy through in-context adaptation alone. Without parameter updates, it progressively adapts to partner behaviors using recent interaction histories, demonstrating few-shot generalization in multi-agent coordination.

5 EXPERIMENTS

5.1 EVALUATION SETUP

Environments. We evaluate our method in the Overcooked environment (Lauffer et al., 2023), a cooperative multi-agent benchmark where two agents prepare and deliver soups under a fixed time limit. Agents can move, wait, or interact with ingredients (e.g., onions, tomatoes) and objects (e.g., pots, plates, dispensers), receiving a sparse reward of 20 for each successful delivery. To assess generality, we test across five layouts—Asymm. Adv., Bothway Coord., Blocked Coord., Coord. Ring, and a multi-recipe variant of it. We evaluate these layouts' performance over 10 episodes per layout, except for the multi-recipe variant, where we use 15 episodes, as we follow the setup from ZSC-Eval (Wang et al., 2024). Each layout introduces distinct coordination challenges through resource distribution and spatial constraints. For example, the Bothway Coord. layout spatially separates work areas, preventing collisions while restricting plate access to a single agent. This asymmetry enforces a role division, with one agent providing plates and the other delivering soups, creating a collision-free coordination scenario. By contrast, the Asymm. Adv. layout distributes resources more evenly across zones, allowing agents to operate largely independently. This balanced structure minimizes the need for close coordination, making it a low-interaction scenario.

We compare CooT against various baselines.

- Behavior cloning (BC) learns to imitate expert demonstrations via supervised learning; it lacks mechanisms to adapt to different partners and is used as a non-aware partner baseline.
- Maximum entropy population training (MEP; Zhao et al., 2023) promotes behavioral diversity by regularizing policies with an entropy objective during population training, improving robustness to unfamiliar partners.
- Hidden-utility self-play (HSP; Yu et al., 2023) induces behavioral diversity by training agents with varied hidden reward functions, improving generalization to unseen partners. Both MEP and HSP are evaluated in dense-reward variants that use reward shaping (e.g., credits for ingredient pickups or placements) to stabilize training. While such shaping is unrealistic in human–AI collaboration, it yields stronger policies and thus provides an upper-bound comparison, whereas CooT is trained only with sparse rewards.
- HSP-fine-tuning (HSP-ft) fine-tunes HSP agents on novel partners to test whether gradient-based adaptation can improve coordination. Learning rates and last-layer updates are varied, following common fine-tuning practices.
- HSP-meta extends HSP with a trajectory encoder inspired by PEARL (Rakelly et al., 2019), an off-policy meta-RL method that learns to quickly adapt. The encoder, optimized with reconstruction loss, encodes recent episodes into a latent context appended to the observation, enabling adaptation to partner behaviors. Since HSP consistently outperforms MEP, this extension provides a stronger and more representative baseline.

Evaluation pipeline. Our evaluation follows ZSC-Eval (Wang et al., 2024), which measures zero-shot coordination with unseen partners. Specifically, evaluation partners are policies trained under diverse hidden reward functions, and a representative subset is selected using Best Response Diversity (the determinant of the similarity matrix of their best responses). Unlike the original protocol, which evaluates across multiple training checkpoints, we use only fully converged checkpoints to remove variability from partially trained behaviors and isolate final performance.

Evaluation metrics. We report Best Response Proximity (BR-prox) and average episode reward. BR-prox (Wang et al., 2024) quantifies how close an agent comes to the ideal best-response performance when paired with a given evaluation partner. More precisely, BR-prox is defined as the ratio between the agent's return when paired with a partner and the return that partner achieves with its own best-response teammate. A higher ratio indicates stronger coordination and generalization. To avoid degenerate cases, we exclude partners whose best-response return is zero. Average episode reward measures overall effectiveness across partners and complements BR-prox. We average both metrics over 50 rollouts per partner.

5.2 COORDINATION PERFORMANCE ACROSS BENCHMARKS

We next present results with unseen policy partners, comparing CooT against strong baselines across diverse Overcooked layouts.

COOT delivers consistently strong results across diverse Overcooked layouts, with its advantage becoming more significant as coordination demands increase. As Table 1 shows, COOT is competitive in Coord. Ring and establishes a clear lead in Multi-recipe, where agents must balance concurrent goals. This suggests that conditioning on past trajectories helps infer partner intentions, enabling robust coordination precisely where baselines break down.

BC performs reasonably in layouts where agents operate in separated zones and collisions are avoided, such as *Asymm. Adv.* and *Bothway Coord.*. However, it struggles once coordination requires navigation or adaptation, as it lacks any mechanism to model or respond to partner behavior.

MEP and HSP perform well in simpler settings, such as *Asymm. Adv.*, a low-interaction scenario requiring little coordination, and the compact *Coord. Ring*, which combines navigation and ingredient handoffs within a constrained space. Yet their reliance on fixed policy pools limits flexibility: they underperform in *Bothway Coord.*, which requires reliable role coordination, and collapse in multi-recipe layouts that demand tracking multiple goals. HSP-ft further highlights the limits of gradient-based adaptation. Its performance is highly sensitive to learning rates and shows little improvement across episodes, sometimes degrading in more complex scenarios (Table 12, Appendix C.3), underscoring the advantage of context-based adaptation.

Table 1: **Benchmark results: CooT outperforms baselines in coordination-heavy layouts.** We report the average episode reward (\uparrow) and BR-prox (\uparrow), both averaged across different layouts. Each entry shows mean \pm std over 50 rollouts. We bold results that lie within the best method's standard deviation range for each layout. CooT maintains strong performance across all settings, with clear advantages in coordination-heavy layouts such as Multi-recipe and Counter Circ.

Layout	Coord.	Ring	Coord. Ring l	Multi-recipe	Counte	r Circ.
	Reward	BR-prox	Reward	BR-prox	Reward	BR-prox
BC	26.24±1.80	0.31±0.02	8.97±0.49	0.10±0.01	10.79±5.33	0.11±0.06
MEP	40.30 ± 3.45	0.47 ± 0.04	16.64 ± 1.16	0.19 ± 0.02	1.89 ± 0.41	0.02 ± 0.00
HSP	41.10 ± 10.03	0.49 ± 0.10	29.35 ± 3.77	0.33 ± 0.04	21.37 ± 2.17	0.23 ± 0.03
HSP-ft	41.30 ± 9.85	0.49 ± 0.10	29.24 ± 3.75	0.33 ± 0.04	21.15 ± 1.53	0.22 ± 0.02
HSP-meta	29.84 ± 3.92	0.35 ± 0.04	30.21 ± 1.37	0.34 ± 0.02	3.28 ± 0.21	0.03 ± 0.00
CooT (Ours)	38.30 ± 3.71	$0.47 {\pm} 0.06$	45.96 ± 3.99	$0.50 {\pm} 0.04$	28.28 ± 2.32	$0.30 {\pm} 0.03$
Layout	Asymm.	Adv.	Bothway	Coord.	Ove	rall
	Reward	BR-prox	Reward	BR-prox	Reward	BR-prox
BC	108.83±6.13	0.53±0.03	98.99±1.30	0.94±0.01	50.76±2.02	0.40 ± 0.02
MEP	127.44 ± 5.66	0.61 ± 0.03	22.76 ± 5.59	0.20 ± 0.05	41.81 ± 0.79	0.30 ± 0.00
HSP	131.78 ± 3.49	0.63 ± 0.02	54.99 ± 3.56	0.53 ± 0.03	55.72 ± 2.48	0.44 ± 0.02
HSP-ft	131.79 ± 3.22	0.63 ± 0.02	55.81 ± 2.71	0.54 ± 0.02	55.86 ± 2.29	0.44 ± 0.02
HSP-meta	113.16 ± 12.72	0.54 ± 0.06	20.44 ± 4.59	0.20 ± 0.05	40.19 ± 2.28	0.29 ± 0.01
CooT (Ours)	129.48 ± 9.34	0.62 ± 0.05	101.93 ± 1.00	0.96 ± 0.01	68.79 ± 2.33	0.57 ± 0.02

Surprisingly, HSP with trajectory conditioning (HSP-meta) underperforms vanilla HSP. Training the trajectory context encoder jointly with the policy introduces challenges: early in learning, noisy or uninformative latents can mislead the policy, and the reconstruction loss may not align with coordination objectives. Subtle partner differences may also be overlooked when reconstructing raw trajectories, limiting the usefulness of the latent context. These results suggest that although trajectory-level conditioning is appealing, effective coordination requires more principled context representations, which are non-trivial to design.

Overall, CooT achieves robust coordination with policy-based evaluation partners. But do these gains transfer to human collaborators, whose behaviors are more varied and less predictable?

5.3 Human-Agent Collaboration Study

Unlike trained agents, humans display diverse styles and adapt unpredictably. To assess whether COOT 's advantages extend beyond controlled benchmarks, we conducted a user study in the *Coord. Ring* layout, which integrates navigation and ingredient coordination within a compact space to elicit diverse collaboration scenarios. A total of 36 participants each played eight rounds, blindly paired through random assignment with one of four representative agents: COOT, HSP, MEP, or BC. We excluded HSP-ft and HSP-meta since preliminary experiments showed they were not competitive and provided little additional insight; restricting the study to four agents also reduced participant fatigue. Each successful delivery contributed 20 points to the score. To ensure consistency, we fixed the time interval between rounds and excluded the top and bottom 10% of participant scores to reduce variability from fatigue or outliers. All participants provided informed consent, and the study was conducted in accordance with standard ethical guidelines. Further details on the human evaluation platform and experimental setup are provided in Appendix D.

Table 2: **Human evaluation returns.** Comparison of mean episode reward (after outlier removal) across methods. CooT achieves the highest return and consistently outperforms baselines, indicating stronger coordination with human partners.

Method	Mean	Std	25%	50%	75%
BC	51.0	3.8	47.5	52.5	55.0
MEP	53.0	5.1	49.4	53.8	57.5
HSP	40.5	5.2	36.9	40.0	45.0
CooT (Ours)	63.5	9.5	56.9	62.5	68.1

Table 3: **Human evaluation ratings.** Human evaluations (1–6 scale, higher is better) for four different agents based on two abilities: Collaborative and Adaptive. Each cell contains the score given by a user. The overall score is the sum of the two individual scores.

Method	Collab.	Adapt.	Overall
BC	2.0	1.8	3.4
MEP	3.4	2.9	6.3
HSP	2.7	2.2	4.9
CooT (Ours)	4.0	3.1	7.1

379

380

381

382

384

385

386

387

388

389

390 391

392

393

394 395

396

397 398 399

400 401

402

403

404 405 406

407 408

409 410

411

412

413

414

415

416

417 418

419

420

421

422 423 424

425 426

427

428 429

430

431

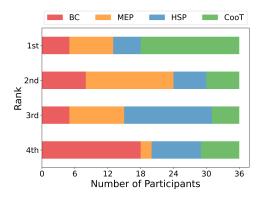
Quantitative results. Table 2 reports average performance across participants after outlier removal. CooT achieved the highest mean score, outperforming all baselines by a large margin. Table 3 summarizes 1-6 ratings of collaboration and adaptivity, where COOT again achieved the best scores on both dimensions. Figure 2 further shows that COOT was most frequently selected as the preferred partner (18 of 36 participants). Together, these results demonstrate that CooT not only delivers the strongest objective outcomes but is also consistently favored in subjective human evaluations.

Qualitative feedback. Post-round questionnaires presented in Table 13 in Appendix D.3 highlight both strengths and limitations of CooT. Participants often noted improvement over time: "Initially it doesn't do well, but gradually the performance improves," and "smoother collaboration over time." Others pointed to supportive behaviors, such as "It collaborates with me well, doing work on its own and helping me with onion preparation." At the same time, some observed occasional blocking or rigidity, e.g., "It blocked me in the beginning, but then it adapted." Overall, COOT was viewed as a capable collaborator, though not always seamless.

Discussion. The human study shows that COOT performs strongly not only with policy-based partners but also with real humans, whose behaviors are more varied and less predictable. Notably, in the Coord. Ring layout where COOT was similar to agent partners, it achieved a clear margin when paired with human participants. These findings suggest that CooT is particularly effective at handling the variability and adaptation demands that arise in real interactions. To further investigate the source of this advantage, the next section analyzes two key properties of COOT: its ability to adapt rapidly to out-of-distribution behaviors and its robustness to non-stationary partners.

ANALYSIS OF ADAPTATION DYNAMICS

The human study suggests that COOT can handle the variability of real interactions, but the mechanisms behind this advantage remain unclear. To probe further, we analyze two key properties of CooT in controlled policy-based settings: its ability to adapt rapidly to new partners and its robustness to sudden changes in partner behavior.



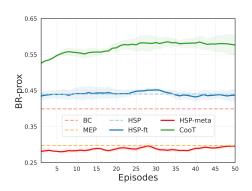


Figure 2: Human study: agent ranking distri- Figure 3: In-context performance improvement **bution.** Number of participants who chose each agent at different rankings. CooT received the highest number of first-place rankings, indicating it is the most preferred collaborator.

of CooT over episodes. As more partner trajectories are observed, CooT steadily improves its coordination strategy, highlighting the advantage of context-based adaptation.

5.4.1 RAPID ADAPTATION THROUGH CONTEXT

We first measure how COOT adapts over time when paired with previously unseen partners. Figure 3 reports averages across seven layouts, each evaluated with 10-15 partners over 50 episodes.

CooT improves markedly within the first 5–8 episodes and continues to refine more gradually up to episode 50, showing that only a handful of trajectories are sufficient to align with a new partner. In contrast, the baseline methods show minimal improvement: HSP-ft exhibits unstable performance with occasional degradation, while HSP-meta demonstrates a rather slow adaptation.

Table 4: Adaptation speed for non-stationary strategies. Episodes needed for CooT to attain its average performance with the new partner.

$P_A \rightarrow P_B$	$P_{15} \rightarrow P_{17}$	$P_{17} \rightarrow P_{15}$	$P_{15} \rightarrow P_{31}$	$P_{31} \rightarrow P_{15}$	$P_{17} \rightarrow P_{31}$	$P_{31} \rightarrow P_{17}$
Episodes to adapt	6	2	6	5	2	1

These results highlight CooT 's ability for rapid context-based adaptation — a crucial property for human–AI collaboration, where effective coordination must be established quickly without prior exposure.

5.4.2 ROBUSTNESS TO NON-STATIONARY PARTNERS

Human partners often change strategies mid-task. To test robustness to such non-stationarity, we conduct a controlled partner-swap experiment: CooT first plays several episodes with one HSP biased partner (P_A) , accumulates context, and then is suddenly paired with a different biased partner (P_B) that follows a distinct preference. We then measure how many episodes it takes for CooT to recover the level of performance it typically achieves with the new partner when trained from scratch. The procedure is repeated in both directions $(P_A \rightarrow P_B \text{ and } P_B \rightarrow P_A)$.

Results in Table 4 show that CooT adapts within at most six episodes, with an average of 3.67 episodes across partner switches. This demonstrates that CooT can rapidly recalibrate its coordination strategy when faced with abrupt partner changes, relying only on its context buffer without fine-tuning or retraining. Such robustness to non-stationary strategies is essential for real-world human–AI collaboration.

Discussion. Together, these results show that CooT can both adapt quickly to new partners and remain stable under abrupt partner changes—two properties that are essential for real-world coordination. As a final check, we perform ablations to verify which design factors contribute most to these capabilities.

5.5 ABLATION STUDIES

To better understand which design choices are most critical for CooT, we perform small-scale ablations focusing on two factors: context length and trajectory shuffling.

Context length. Table 9 in Appendix C.1 shows that performance improves when conditioning on longer contexts, up to five episodes. Beyond this point, gains diminish while computation increases, so we adopt five episodes as the default.

Trajectory shuffling. Table 11 in Appendix C.2 shows that during training, step-wise shuffling disrupts temporal structure and harms performance, while chunk-wise shuffling has little effect. This confirms that preserving long-range dependencies in interaction histories is important.

6 Conclusion

We introduced Coordination Transformer (CooT ②), an in-context coordination framework that enables agents to adapt to unseen partners by conditioning on recent interactions. Unlike prior approaches emphasizing task generalization or large training populations, CooT targets partner generalization, predicting best-response actions without fine-tuning. Experiments on Overcooked show that CooT outperforms strong baselines with both evaluation agents and humans, while analyses highlight its rapid adaptation and robustness to unseen partners. Our main contributions are the formulation of partner-centric in-context coordination and extensive empirical validation in a challenging coordination domain. However, our work is limited by evaluation in a relatively structured environment and the lack of explicit modeling for mutual adaptation. Future work can address these limitations by extending CooT to more complex, co-adaptive, and real-world multi-agent settings.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, the Supplementary Material includes the training and evaluation scripts, together with configuration files and detailed instructions for reproducing the reported results.

BIBLIOGRAPHY

- Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. In *Mathematics of operations research*, 2002.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Neural Information Processing Systems*, 2020.
- Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9, 2012.
- 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 *Neural Information Processing Systems*, 2019.
- Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. Generating diverse cooperative agents by learning incompatible policies. In *International Conference on Learning Representations*, 2023.
- Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Neural Information Processing Systems*, 2021.
- Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl²: Fast reinforcement learning via slow reinforcement learning. *arXiv* preprint arXiv:1611.02779, 2016.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.
- Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. In *International Conference on Learning Representations*, 2017.
- Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, 2020.
- Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In *International Conference on Machine Learning*, 2021.
- Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.
- Yuheng Jing, Kai Li, Bingyun Liu, Yifan Zang, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Towards offline opponent modeling with in-context learning. In *International Conference on Learning Representations*, 2023.
- Louis Kirsch, James Harrison, C. Freeman, Jascha Sohl-Dickstein, and Jürgen Schmidhuber. Towards general-purpose in-context learning agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.
- Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 2012.
- Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, maxime gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations*, 2023.
- Niklas Lauffer, Ameesh Shah, Micah Carroll, Michael D Dennis, and Stuart Russell. Who needs to know? minimal knowledge for optimal coordination. In *International Conference on Machine Learning*, 2023.

543

544

546 547

548

549

550

551

552

553 554

555

556

558

559

560 561

562

563

564

565

566 567

568

569

570

571

572

573 574

575 576

577

578

579

580

581 582

583

584 585

586

588

589

590

591

- 540 Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In Neural Informa-542 tion Processing Systems, 2023.
 - Yang Li, Shao Zhang, Jichen Sun, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Cooperative openended learning framework for zero-shot coordination. In International Conference on Machine Learning, 2023a.
 - Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In International Conference on Machine Learning, 2023b.
 - Qian Long*, Zihan Zhou*, Abhinav Gupta, Fei Fang, Yi Wu†, and Xiaolong Wang†. Evolutionary population curriculum for scaling multi-agent reinforcement learning. In International Conference on Learning Representations, 2020.
 - Xingzhou Lou, Jiaxian Guo, Junge Zhang, Jun Wang, Kaiqi Huang, and Yali Du. Pecan: Leveraging policy ensemble for context-aware zero-shot human-ai coordination. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, 2023.
 - Keane Lucas and Ross E Allen. Any-play: An intrinsic augmentation for zero-shot coordination. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, 2022.
 - Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, 2021.
 - Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. The Knowledge Engineering Review, 27, 2012.
 - 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, 2023.
 - Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In International Conference on Machine Learning, 2019.
 - John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.
 - Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In Proceedings of the AAAI Conference on Artificial Intelligence, 2010.
 - DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In Neural Information Processing Systems, 2021.
 - Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. Neural computation, 6, 1994.
 - Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017.
 - Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In International Conference on Machine Learning, 2023.
 - Xihuai Wang, Zheng Tian, Ziyu Wan, Ying Wen, Jun Wang, and Weinan Zhang. Order matters: Agent-by-agent policy optimization. In *International Conference on Learning Representations*, 2023.

- Xihuai Wang, Shao Zhang, Wenhao Zhang, Wentao Dong, Jingxiao Chen, Ying Wen, and Weinan Zhang. Zsc-eval: An evaluation toolkit and benchmark for multi-agent zero-shot. In *Neural Information Processing Systems*, 2024.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Neural Information Processing Systems*, 2022.
- Jinwei Xing, Takashi Nagata, Kexin Chen, Xinyun Zou, Emre Neftci, and Jeffrey L Krichmar. Domain adaptation in reinforcement learning via latent unified state representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- Ke Xue, Yutong Wang, Cong Guan, Lei Yuan, Haobo Fu, Qiang Fu, Chao Qian, and Yang Yu. Heterogeneous multi-agent zero-shot coordination by coevolution. *IEEE Transactions on Evolutionary Computation*, 2024.
- Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10, 2013.
- 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 *Neural Information Processing Systems*, 2022a.
- Chao Yu, Jiaxuan Gao, Weilin Liu, Botian Xu, Hao Tang, Jiaqi Yang, Yu Wang, and Yi Wu. Learning zero-shot cooperation with humans, assuming humans are biased. In *International Conference on Learning Representations*, 2023.
- Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for contentrich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022b.
- Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv* preprint arXiv:2006.10742, 2020.
- Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.
- 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*, 2023.
- Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: variational bayes-adaptive deep rl via meta-learning. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

APPENDIX

Table of Contents

A	Experiment Environment	13
В	Implementation and Dataset Details	14
	B.1 Source and Licensing of Policy Pools and Datasets	14
	B.2 CooT Training Details	15
	B.3 Baseline and Training Implementation	15
	B.4 Hardware Specifications and Training Time	17
	B.5 Inference Time Comparison	17
C	Complementary Experiments	17
	C.1 Context Length	17
	C.2 Context Shuffling	17
	C.3 Fine-Tuning	18
	C.4 Adaptation	18
D	Additional Contexts for Human Experiment	19
	D.1 Experiment Setup	19
	D.2 Experiment Platform	20
	D.3 Qualitative Feedback from Participants	21
E	Algorithm	24
F	The Use of Large Language Models	25

A EXPERIMENT ENVIRONMENT

We evaluted in the Overcooked environment (Carroll et al., 2019) inplemented in ZSC-Eval (Wang et al., 2024)¹. We used seven layouts, including Forced Coordination (*Forced Coord.*), Bothway Coordination (*Bothway Coord.*), Coordination Ring (*Coord. Ring*), Counter Circuit (*Counter Circ.*), Asymmetric Advantages (*Asymm. Adv.*), Forced Coordination with Multi-recipe (*Forced Coor. multi-recipe*), and Coordination Ring with Multi-Recipe (*Coord. Ring multi-recipe*). The multi-recipe layouts have onion (O) and tomatoes (T) as ingredients, which expands the range of recipes from just onion soup (3O) to five types of soups, including mix soup (1O1T), less onion soup (2O), tomato-onion soup (2T1O), onion-tomato soup (2O1T), and onion soup (3O). The following are the details and main challenges for each layout.

Bothway Coordination. In this variant, both players can access onions and pots, which broadens the range of feasible strategies and introduces new opportunities for cooperation. This layout helps reduce idle time seen in the Forced Coordination setting and encourages more diverse policies. Still, since plates and the serving station remain confined to one side, effective teamwork is essential to fulfill orders.

Coordination Ring. The Coordination Ring features a compact, circular layout that facilitates close agent interaction. Ingredients, plates, and the serving station are grouped in the bottom-left area, while cooking pots are placed in the top-right. This spatial arrangement drives continuous movement and requires players to coordinate effectively as they manage shared resources and navigate the kitchen.

Counter Circuit. Similar in shape to Coordination Ring, the Counter Circuit introduces a central, elongated table, creating narrow pathways that often cause congestion. This layout demands careful movement planning, as agents must avoid blocking one another. A common cooperative tactic involves staging onions in the center to streamline ingredient transfer.

https://github.com/sjtu-marl/ZSC-Eval, with MIT License.

Asymmetric Advantages. This layout divides the kitchen into two largely self-contained workspaces while maintaining interdependence through asymmetrically shared resources. Each player has unique access to ingredients and serving stations, while two centrally located pots are jointly accessible. Notably, one player benefits from closer proximity to the serving station, encouraging the development of collaboration strategies to balance workload and improve efficiency.

Coordination Ring with Multi-Recipe. This extended version of Coordination Ring includes tomatoes positioned near the serving area in the bottom-left corner, increasing task complexity. The added ingredient and recipe variety heighten the need for coordinated planning and amplify the importance of cooperation in fulfilling diverse orders.

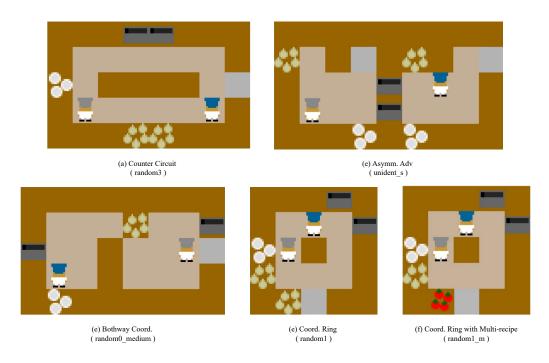


Figure 4: Used layouts in Overcooked.

B IMPLEMENTATION AND DATASET DETAILS

B.1 SOURCE AND LICENSING OF POLICY POOLS AND DATASETS

HSP and MEP training datasets. In our setting, we adopt behavior-preferring rewards, which refer to event-based biased reward functions. These rewards assign credit to specific events—such as picking up an onion or placing an ingredient into the pot. We assume that the behavioral requirements of deployment-time partners can be effectively modeled using this set of reward functions. For details on the reward specifications and policy training under these settings, we refer readers to ZSC-eval (Wang et al., 2024).

For two-stage algorithms such as HSP (Yu et al., 2023) and MEP (Zhao et al., 2023), while we do not train these policies ourselves, we construct the candidate policy pool by loading pretrained checkpoints from the ZSC-eval repository, which contains a diverse collection of agents trained under different bias reward settings and algorithmic configurations. These policies exhibit varied behaviors and serve as potential partners for training adaptive agents in the second stage. Following the settings in ZSC-eval (Wang et al., 2024), we selected 36 candidate policies as the basis for partner selection.

COOT and BC training datasets. We construct our expert dataset mainly following the same procedure as in HSP. To build a robust and diverse partner pool Π_{train} for training adaptive policies, we include 36 agents sourced from both MEP and HSP. Specifically, we collect 15 MEP agents with

final skill levels, and 21 HSP agents greedily selected based on an event-based diversity score d_i , which measures the expected frequency of key events.

For each selected pair $(\pi_i^{\rm p},\pi_i^{\rm br})\in\Pi_{\rm train}$, we collect J joint trajectories τ , where J=200 for MEP agents and J=250 for HSP agents (including 220 from final-skill-level and 30 from mid-skill-level checkpoints). While CooT utilizes the current state s_t and the context C to predict \hat{a} , the BC baseline is trained using the same dataset by conditioning only on the state s_t . Further details of implementation and training will be discussed in the upcoming paragraphs.

B.2 COOT TRAINING DETAILS

Generation of training dataset. To train our Coordination Transformer (CooT), we begin with a set of pretrained policy pairs consisting of biased partner agents and their corresponding best-response policies. For each policy pair, we first select one pair and sample T episodes of rollouts, which we use to construct a context C. This process is repeated K times per policy pair to generate K distinct contexts. For each context C, we then sample L different query states s_h from the rollout trajectories. Each query state, combined with its associated context C and the corresponding optimal action \hat{a} , forms a single training data point of the form (s_h, C, \hat{a}) . Given a total of M policy pairs, this procedure results in a dataset containing $M \times K \times L$ training examples. More detailed information can be found in Table 5.

Online deployment. During online deployment, CooT operates with a dynamically updated context C to enable in-context adaptation. At the start of the first episode, C is initialized as an empty context. Since no prior trajectory is available, CooT selects actions based solely on the current state and the empty context. After completing the first episode, the entire trajectory is stored and used to construct a new context, which is then appended to C. As a result, C contains one populated trajectory and T-1 empty slots (assuming a fixed context length of T). In subsequent episodes, CooT utilizes the current buffer to condition its action predictions, allowing it to adapt based on accumulated experience. The context is managed using a first-in, first-out (FIFO) policy, ensuring that the most recent T trajectories are always retained. This procedure is repeated iteratively throughout the evaluation phase to simulate continual adaptation in a coordination setting. The main method details are provided in Algorithm 1 and Algorithm 2. Algorithm 1 outlines the training process of CooT, while Algorithm 2 outlines the evaluation partners are selected.

B.3 BASELINE AND TRAINING IMPLEMENTATION

Our baseline codebase is primarily based on two open-source frameworks: Imitation², an imitation learning library built on Stable Baselines3³, and ZSC-eval (Wang et al., 2024), a benchmark codebase for Zero-Shot Coordination (ZSC). We use the original implementation of Behavioral Cloning (BC) from Imitation, while the implementations and hyperparameter settings of HSP and MEP are directly inherited from ZSC-eval and HSP (Yu et al., 2023).

All models are evaluated in sparse-reward settings under ZSC environments to better reflect real deployment. To provide a more comprehensive comparison, we additionally report the performance of HSP and MEP trained with dense rewards, which offer frequent intermediate feedback (e.g., picking up onions, placing them in pots) to facilitate more stable and efficient training.

²https://github.com/HumanCompatibleAI/imitation, with MIT license

³https://github.com/DLR-RM/stable-baselines3, with MIT license

Dataset	Parameter	Value
	Number of partner policy pairs (M)	36
	Number of contexts per pair (K)	125
CooT Dataset	Number of rollouts per context (T)	5
	Number of data per context (L)	70
	Total dataset size $(M \times K \times L)$	315,000

Table 5: **Dataset configuration for training the Coordination Transformer (CooT).** It specifies the number of partner policy pairs, contexts, rollouts, and resulting total dataset size.

Method	Hyperparameter	Value
	Batch size	256
	Learning rate	0.001
	Optimizer	Adam
ВС	Scheduler	CosineAnnealingLR
ьс	Max Epochs	50
	Early Stopping Patience	5
	Validation Split	0.1
	Scheduler η_{\min}	$\frac{lr}{100} = 3e-6$
	Batch size	120
	Learning rate	5e-5
	Optimizer	Adam
	Scheduler	LambdaLR
	Weight decay	1e-3
	Dropout	0.3
CooT	Gradient clip norm	0.25
C001	Model	GPT-2
	Hidden layers	4
	Attention heads	2
	Embedding size	128
	Max Epochs	70
	Early Stopping Patience	25
	Entropy coefficient	0.01
	Gradient clip norm	10.0
	GAE lambda	0.95
	Discount factor (γ)	0.99
	Value loss	Huber loss
	Huber delta	10.0
Stage2 of HSP and MEP	Optimizer	Adam
Stage2 of 1151 and WEI	Optimizer epsilon	1e-5
	Learning rate	5e-4
	Parallel environment threads	100
	Environment steps	10M
	Episode length	200
	Reward shaping horizon	10M
	Policy pool size	36
<u> </u>	Parallel environment threads	50
HSP-meta	Encoder loss	Reconstruction loss
1151 meta	Encoder layers	3
	Encoder lr	1e-5

Table 6: **Training hyperparameters for all methods.** We list model settings for CooT, PPO parameters for HSP/MEP, and additional encoder parameters for HSP-meta.

Table 7: **Computational resources used.** Specifications of the four workstations used for training and evaluation, including CPU, GPU, and memory.

Workstation	CPU	GPU	RAM
Workstation 1	Intel Xeon W-2255	NVIDIA GeForce RTX 3080	125 GiB
Workstation 2	Intel Xeon W-2255	NVIDIA GeForce RTX 3090 ×2	125 GiB
Workstation 3	Intel Xeon w7-2475X	NVIDIA GeForce RTX 4090	125 GiB
Workstation 4	Intel Xeon w7-2475X	NVIDIA GeForce RTX 4090 ×2	125 GiB

B.4 HARDWARE SPECIFICATIONS AND TRAINING TIME

We performed the experiments using the workstations listed in Table 7. Our method takes approximately 19 hours to train for each layout. Reproducing all results, including the baselines, requires approximately 300 GPU hours when run sequentially.

Table 8: **Average per-step inference time across model architectures.** Reported times reflect pure model forward pass per action, excluding environment interaction.

	CooT	MEP	HSP	BC
Inference Time (ms)	2.41	1.73	1.77	0.54

B.5 INFERENCE TIME COMPARISON

To complement the training cost reported above, we provide a comparison of pure model inference time, excluding environment interaction, across the architectures used in our experiments. Table 8 reports the average per-step inference time. While transformer-based ICL introduces additional overhead compared to lightweight baselines, the runtime remains practical in our multi-agent setting, as CooT requires only 2.41 ms per step, which is acceptable for our simulation-based evaluations.

C COMPLEMENTARY EXPERIMENTS

C.1 CONTEXT LENGTH

Performance improves steadily as context length increases up to 5 episodes, reaching the best mean reward of 41.1. Beyond this point, gains diminish and even drop slightly at 7 episodes, indicating that overly long contexts may dilute relevant information while adding computational overhead. We therefore adopt 5 episodes as the default setting, balancing effectiveness and efficiency.

C.2 CONTEXT SHUFFLING

While CooT demonstrates robust performance in unseen partner coordination, we investigate whether its generalization ability can be further enhanced through improved training-time augmentation strategies. Specifically, we explore whether trajectory augmentation can help the model acquire more robust coordination behaviors. Motivated by Decision Pretrained Transformers (DPT) (Lee et al., 2023), where step-wise trajectory shuffling enhanced generalization in partially observable navigation tasks, we examine whether similar augmentation methods benefit our multi-agent coordination setting.

To this end, we evaluate two trajectory-shuffling strategies, step-wise and chunk-wise, in the *Coord. Ring* layout over 50 rollouts each. Although step-wise shuffling has shown benefits in simple navigation environments such as Dark-Room (Zintgraf et al., 2021), where agents operate under short horizons and limited observability, we observe a performance decline when applying this strategy in Overcooked. Unlike Dark-Room's short-horizon navigation tasks, Overcooked requires tight coordination over extended horizons. In such settings, preserving temporal structure remains critical,

Table 9: **Effect of context length on coordination performance in** *Coord. Ring.* Performance improves steadily as context length increases from 1 to 5 episodes, but gains diminish when extended to 7 episodes.

5 episodes

7 episodes

ended to 7 episodes.				
Context length	Reward (†)	-	_	
1 episode	33.87	•		
3 episodes	34.36			

41.11

36.04

Table 10: **Average episode reward and BRprox under different shuffle strategies on** *Co***ord. Ring.** Preserving temporal structure results in better coordination and improved adaptation performance.

Shuffle strategy	Reward (†)	BR-prox (↑)
No Shuffling	37.47±5.67	0.44 ± 0.094
Step-wise	29.41 ± 5.23	0.36 ± 0.084
Chunk-wise	38.30 ± 3.71	$0.47 {\pm} 0.056$

Table 11: Average episode reward and BR-prox under different shuffle strategies on *Coord. Ring*. Preserving temporal structure results in better coordination and improved adaptation performance.

Shuffle strategy	Reward (†)	BR-prox (†)
No Shuffling	37.47 ± 5.67	$0.44{\pm}0.094$
Step-wise	29.41 ± 5.23	0.36 ± 0.084
Chunk-wise	38.30 ± 3.71	$0.47 {\pm} 0.056$

as effective coordination in Overcooked requires agents to act in a tightly timed relation to one another to avoid unnecessary delays or idle time.

As shown in Table 11, chunk-wise permutation, which shuffles multi-step segments while preserving key temporal dependencies, shows a slight improvement over step-wise shuffling and no augmentation. Interestingly, step-wise shuffling disrupts essential temporal continuity despite increasing data diversity and performs worse than unaugmented data. These findings emphasize the importance of preserving long-range temporal structure in trajectory augmentation, especially for multi-agent tasks that rely on temporally extended interactions.

C.3 FINE-TUNING

These results illustrate the brittleness of gradient-based adaptation in coordination settings. While large learning rates (e.g., 10^{-4}) consistently destabilized training, leading to sharp performance drops, smaller learning rates (10^{-7} and 10^{-8}) yielded negligible improvement. Partial fine-tuning (p-ft) often underperformed full updates, suggesting that restricting adaptation to the actor head fails to capture the dynamics needed for partner alignment. Overall, the limited and unstable gains from fine-tuning highlight the contrast with CooT, which adapts stably and rapidly through in-context updates without parameter changes.

C.4 ADAPTATION

Figure 5 further highlights the performance gap between COOT and the HSP-Meta baseline. Across all five layouts, COOT generally improves with additional episodes, while HSP-Meta remains largely flat. In *Coord. Ring*, COOT rises from initial BR-prox values around 0.4 to over 0.5, The difference is most striking in the *Coord. Ring Multi-recipe* layout: COOT continues to improve over the full 50 episodes, yet HSP-Meta stagnates early and even falls below the HSP baseline.

These results suggest that the trajectory encoder introduced in HSP-Meta does not produce useful latent representations for coordination. One likely reason is that the reconstruction loss does not align with coordination objectives: subtle behavioral cues are either smoothed out or misrepresented, leaving the policy unable to differentiate between partners. By contrast, CooT conditions directly on observed interactions and learns to map them to best-response actions, leading to consistent adaptation across all layouts.

Table 12: Comparison of online fine-tuning baselines (MAPPO with different learning rates) against CooT. Each model was evaluated with 50-episode interactions across 9 partner seeds. "p-ft" denotes updating only the actor head. Fine-tuning remains unstable: large learning rates cause collapse, while small rates yield no consistent gains.

Partner	no ft	1e-8	1e-7	1e-6	1e-4	1e-8 p-ft	1e-7 p-ft	1e-6 p-ft
2	8.0	7.6	6.8	6.4	1.6	8.0	8.0	6.8
12	6.8	6.8	7.2	8.0	5.2	6.8	6.8	6.4
15	39.2	38.4	39.6	41.6	32.0	39.2	39.2	38.4
16	36.8	38.4	38.4	37.2	32.4	36.8	36.8	36.0
17	28.4	26.4	30.4	28.4	16.8	28.4	28.0	28.4
20	41.2	40.8	44.0	38.0	31.2	41.2	41.2	40.0
27	37.2	37.6	35.6	32.4	32.4	37.2	38.0	38.8
31	47.6	47.6	46.4	43.2	31.6	47.6	47.6	46.4
50	21.2	20.8	20.8	19.6	15.6	21.2	20.8	20.8
Avg.	29.6	29.4	29.9	28.3	22.1	29.6	29.6	29.1

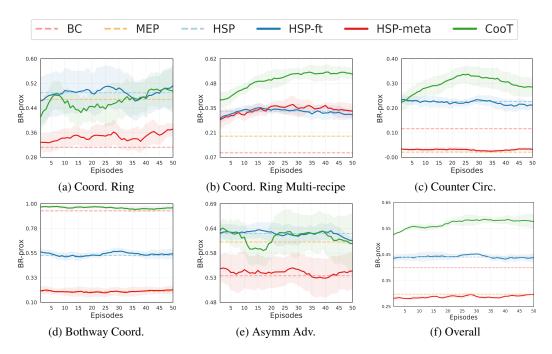


Figure 5: CooT performance across layouts. Learning curves on six evaluation layouts: Coord. Ring, Coord. Ring Multi-recipe, Counter Circ., Bothway Coord., Asymm Adv., and the aggregate result across all layouts (Overall).

D ADDITIONAL CONTEXTS FOR HUMAN EXPERIMENT

D.1 EXPERIMENT SETUP

We recruited and verified 36 participants, with a gender distribution of 27 males and 9 females, for the human experiment. Seven participants have prior experience in playing the actual Overcooked!. To mitigate learning effects among the subjects, the order of the agents was randomized. The participants are required to play 200 per episode, 1600 timesteps in total, with 8 episodes for each agent (approximately 5.3 minutes). This leads to a total time of around 30 minutes. The names of the algorithms used by the agents were not visible during the experiments. Agents were differentiated

solely by color. Participants were asked to rank the agents after each round, and their trajectories were recorded. All data collection was conducted with the consent of the participants.

D.2 EXPERIMENT PLATFORM

We built our human evaluation platform on top of the ZSC-Eval benchmark (Wang et al., 2024)^A, which provides a standardized environment for testing human-AI coordination in Overcooked. To adapt it to our setting, we modified the system to support repeated interactions with the same agent, enabling context accumulation over multiple episodes.

During the experiment, participants controlled one character using keyboard inputs, while the partner was controlled by one of the four agents under evaluation: CooT, MEP, HSP, or BC. To reduce potential bias, agent identities were hidden and replaced with randomized colors. The interface presented real-time feedback, and all trajectories were automatically recorded for later analysis. Figures 6 to 9 show the platform's interface and experiment flow.

Table 13: **Representative participant feedback on different agents.** Five comments were selected for each method following the criteria described in the main text.

Method	Aspect	Quote
ВС	Negative feed- back	"Is he stupid or what?"
BC	Lack of adaptation	"Never adapts."
BC	Blocking behavior	"It kept blocking my way and not taking the soup."
BC	Slight improve- ment	"Initially didn't know what I want. But improved, still too late."
BC	Independence only	"Can play solo but ignores me."
MEP	Positive impression	"The agent understands their job and collaborate well with me."
MEP	Blocking behavior	"Blocks my way; doesn't know what it was doing sometimes."
MEP	Partial adapta- tion	"Tried to adapt but too late."
MEP	Independent play	"Can do task independently."
MEP	Limitation	"Didn't adapt at all."
HSP	Partial adaptation	"I think it improves over time."
HSP	Blocking behavior	"Blocks my way too often."
HSP	Positive impression	"Good, but lacked speed."
HSP	Mixed feed- back	"Sometimes adapts, sometimes fails."
HSP	Limitation	"IDLE a lot especially after I deliver the soup."
CooT	Positive impression	"It collaborates with me well, doing work on its own and helpi with onion preparation."
CooT	Adaptation	"Initially it doesn't do well, but gradually the performance impr
CooT	Blocking then adapts	"It blocked me in the beginning, but then it adapted."

CooT Smooth collaboration "Smoother collaboration over time."

CooT Limitation "A stubborn model. S/he just did the same strategy every time and didn't help me at all."

D.3 QUALITATIVE FEEDBACK FROM PARTICIPANTS

To complement the quantitative rankings, we also collected qualitative feedback from participants after each interaction session. These open-ended comments provide insight into subjective impressions of each agent's behavior, including adaptiveness, blocking tendencies, and responsiveness. We organize these comments in Table 13, aligned with the BC, MEP, HSP, COOT order based on participant rankings.

Layout	Game Length (sec)	Game number	Agent ID	
random1	400	3 of 4	3 of 4	

When playing, please make sure to pay attention to the <u>hat colors</u> of your partners! You will be asked to rank the agents (identified by their hat colors) according to their performance.

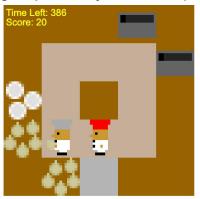


Figure 6: **Main experiment layout for human study.** *Coord. Ring* is chosen since it requires both navigation and ingredient coordination. Thus, it can observe more coordination situations.

Questionnaire

Please rank the agents by dragging the corresponding figures based on your feelings of the agents' cooperation ability. Please rank the agents from best to worst, from top to bottom.



Figure 7: **Human subjective perception ranking system.** Interface used in the user study for collecting participants' subjective rankings of agents. After each round, participants were asked to compare all agents they interacted with and select which partner they preferred most. This ranking procedure complements quantitative metrics by capturing human impressions of collaboration quality.

Experimental Statement 1. Purpose You have been asked to participate in a research study that studies human-Al coordination. We would like your permission to enroll you as a participant in this research study. The instruments involved in the experiment are a computer screen and a keyboard. The experimental task consisted of playing the computer game Overcooked and manipulating the keyboard to coordinate with the AI agent to cook and serve dishes 2. Procedure In this study, you should read the experimental instructions and ensure that you understand the experimental content. The whole experiment process lasts about 30 minutes, and the experiment is divided into the following steps: (1) Read and sign the experimental statement, and you need to fill in a guestionnaire; (2) Test the experimental instrument, and adjust the seat height, sitting posture, and the distance between your eyes and the screen. Please ensure that you are in a comfortable sitting position during the experiment; (3) You will first try out the game actions you learned in the tutorial within a simple layout to familiarize yourself with the game mechanics; (4) Start the formal experiment. Please cooperate with the AI agents to get as much scores as possible. You will play with 4 agents in 1 layout. You need to rank the performance of these four agents. After each round, we will ask you to add the current agent to the ranking. After the game ends in each layout, we need to confirm your ranking of the agents. 3. Risks and Discomforts The only potential risk factor for this experiment is trace electron radiation from the computer. Relevant studies have shown that radiation from computers and related peripherals will not cause harm to the human body. 4. Compensation Each participant who completes the experiment will be paid around 6~7 USD. 5. Confidentiality The results of this study may be published in an academic journal/book or used for teaching purposes. However, your name or other identifiers will not be used in any publication or teaching materials without your specific permission. In addition, if photographs, audio tapes or videotapes were taken during the study that would identify you, then you must give special permission for their use I confirm that the purpose of the research, the study procedures and the possible risks and discomforts as well as potential benefits that I may experience have been explained to me. All my questions have been satisfactorily answered. I have read this consent form. Clicking the button below indicates my willingness to participate in this study. Figure 8: Statements for human study. Consent and instruction form provided to participants before the experiment. It outlines the purpose of the study (human–AI coordination in Overcooked), the procedure (tutorial, gameplay with four agents, and post-round rankings), potential risks and discomforts, compensation details, and confidentiality terms. This ensured that participants were fully informed and agreed to the study protocol before beginning the human-agent collaboration tasks.

Instructions Please read the following instructions carefully. In this task, you will play in a cooking game as one of the two chefs in Movement and interactions a restaurant that serves onion soup. The chef in you control wearing a gray hat. One of the game layouts looks like: You can move up, down, left, and right using the arrow keys, and interact with objects using the spacebar. You can interact with objects by facing them and pressing space bar. Note that you and your partner cannot occupy the same location. Cooking There are a number of objects in the game, labeled here: Cooking Soup Once 3 onions are in the pot, the soup begins to cook. After the timer gets to 20, the soup will be ready to be served. To serve the soup, bring a dish over and interact with the pot. Goal Your goal in this task is to serve as many of the orders as you can before each level ends. The current score and time left for you are shown in the upper left of game.

After clicking "Start Playing", you will first play in a warmup trial, where scores will not be recorded.

After the warmup trial, the official experiments will be conducted in 3 layouts. You will complete 7 games with 7 different agents in each layout.

When playing, please make sure to pay attention to the <u>hat colors</u> of your partners! You will be asked to rank the agents (identified by their hat colors) according to their performance.

Figure 9: **Instructions for human study.** Guidelines were presented to participants before starting the Overcooked sessions. The instructions described the objective of the task, the number of episodes to be played, and the anonymity of the partner agents (displayed only by color). This ensured participants understood the procedure while minimizing bias toward specific algorithms.

ALGORITHM

1244 1245 1246

1247

1242

1243

```
Algorithm 1 Agent Pool Construction and Training
 1: // Generating agent pool
2: Initialize empty pool \Pi_0
```

```
1248
1249
           3: for i in P do
                  Sample hidden reward function r_i^w from reward space R
1250
                 Train \pi_i^p and its best response \pi_i^{br} using PPO
1251
                 Add (\pi_i^p, \pi_i^{br}) to \Pi_0
1252
           7: end for
1253
           8: // Partner selection for training
1254
           9: Initialize empty training pool \Pi_{train}
          10: for (\pi_i^p, \pi_i^{br}) in \Pi_0 do
1255
1256
                  Rollout trajectories and Compute event-based diversity d_i of \pi_i^{bi}
```

12: **end for** 1257 13: Select top-M agents with highest d_i values as S1258

14: Add corresponding agents to Π_{train} 15: Remove corresponding agenst from Π_0 1260 16: // Construct training dataset 1261 17: Initialize empty dataset D

18: **for** (π_j^p, π_j^{br}) in Π_{train} **do** 19: **for** k in K **do** 1262 1263 20: Rollout T trajectories as context C1264 for l in L do 21: 1265 22: Sample query state $s_h \sim C$ 23: Let $a^* = \pi_i^{\text{br}}(s_h)$ 1266

24: Add data (s_h, C, a^*) to D1267 25: end for 1268 26: end for 1269 **27: end for** 1270 28: // Model training

29: Initialize model M_{θ} 1271 30: while not converged do 1272 31: Sample (s_h, τ_j, a^*) from D 1273

32: Predict action distribution $\hat{p} = M_{\theta}(\cdot \mid s_h, \tau_j)$ 1274 33: Compute loss \mathcal{L} given \hat{p} and Update θ 1275

34: end while

1280

Algorithm 2 Evaluation and Online Deployment (Wang et al., 2024)

```
1281
           1: // Evaluation partner selection
           2: for (\pi_i^p, \pi_i^{br}) in \Pi_0 do
1282
           3:
                  Rollout trajectories \tau_i
1283
                  Embed features of \tau_i into \phi_i
1284
           5: end for
1285
           6: Compute similarity matrix K from \{\phi_i\}_{i=1}^K
1286
           7: Sample subset S from top-N candidates of K
           8: Define evaluation set \Pi_{\text{eval}} = \{\pi_s^p\}_{s \in \mathcal{S}}
1287
           9: // Online deployment
1288
          10: Sample unseen partner \pi_s^p \sim \Pi_{\text{eval}}
1289
          11: Initialize fixed-length context C = \{\}
1290
          12: for episode = 1 to E do
1291
                  for timestep t = 1 to Z do
          13:
1292
          14:
                     Observe s_t, predict a_t \sim M_{\theta}(\cdot \mid s_t, C)
          15:
                     Execute a_t with partner, observe (s_t, a_t, r_t)
1293
          16:
                  end for
1294
          17:
                  Append episode trajectory to context C
1295
          18: end for
```

F THE USE OF LARGE LANGUAGE MODELS

We used large language models (LLMs) in limited ways that did not affect the scientific contributions of this work. Specifically, LLMs were employed to (1) polish and improve the clarity of writing without altering the technical content, (2) help organize and summarize qualitative feedback collected from human study participants, and (3) assist in designing and refining figures for presentation purposes. All conceptual, methodological, and analytical contributions, including study design, data analysis, and interpretation of results, were carried out solely by the authors.