AUTONOMY, ARTIFICIAL INTELLIGENCE & ROBOTICS

Adaptive Adversarial Autonomy: Scaling Stackelberg Equilibria to Continuous Ground Combat Domains via Latent-Context Hypernetworks

Bulent Soykan, Ghaith Rabadi, Grace Bochenek, and Victor J. Paul

GVSETS GROUND VEHICLE SYSTEMS ENGINEERING & TECHNOLOGY SYMPOSIUM
NDIA Michigan Chapter

Outline

1. Introduction & Motivation
2. Background & Related Work
3. Methodology
4. Experimental Setup
5. Results & Analysis
6. Conclusion & Future Work

The Automated Test & Evaluation Challenge

Motivation: Deploying RCVs in Future Combat Formations

- Commercial autonomous vehicles operate in structured, cooperative environments.
- **Military Robotic Combat Vehicles (RCVs)** operate in adversarial domains where the environment itself is often weaponized.
- Adversaries actively shape the battlefield: utilizing terrain, placing mines, establishing kill zones to exploit predictable Blue autonomy.

The Gap in Simulation-Based Training

- Current simulations rely on scripted Enemy AI, static behavior trees, or finite state machines.
- Agents trained against static adversaries suffer from *catastrophic distribution shift* when facing novel tactics (i.e., they overfit).
- **Requirement:** "Adversarial Digital Twins"—automated opposing forces that continuously evolve to exploit weaknesses prior to physical fielding.

Stackelberg Equilibria & MARL Limitations

Stackelberg Games in RL

- Hierarchical decision-making: A **Leader** (Red) commits to a strategy, and a **Follower** (Blue) optimizes a policy in response.
- Modeled as a Leader-POMDP where the Leader's policy is a hidden state variable.

The Scalability Bottleneck

- Prior state-of-the-art (e.g., the STACKELBERG framework) relies on *explicit state enumeration* (Observed Queries).
- The Follower must ask the Leader for its action in every possible state.
- **Problem:** Computationally intractable in continuous domains like terrain-based maneuver where the state space is infinite.

PSRO & Context-Aware Meta-Learning

Policy Space Response Oracles (PSRO)

- Prevents strategy cycling and over-fitting by maintaining a population of policies.
- Iteratively adds an approximate best response to the empirical meta-strategy of the opponent population.
- Forces the Blue agent to learn a generalized counter-strategy across a population of tactical possibilities (e.g., flanking, area denial).

Context-Aware RL & Hypernetworks

- Standard Context-Aware RL concatenates context z with the state s. Yields a *weak* inductive bias for adaptation.
- **Hypernetworks:** One neural network generates the weights of another. Allows an agent to rapidly adapt its entire behavior mapping based on a context embedding.

Proposed Architecture Overview

Latent-Context Hypernetwork Stackelberg PSRO

To solve for Stackelberg equilibria in high-dimensional continuous settings, we decompose the problem into three coupled components:

1. **Latent Probe Wrapper:** An implicit query mechanism gathering information via short probing trajectories.
2. **Transformer Context Encoder:** Compresses probing trajectories into a latent strategy embedding.
3. **Hypernetwork Policy:** Adapts the agent's control logic (neural network weights) in real-time based on the encoded intent.

Problem Formulation: Continuous Leader-Follower Game

- **Leader (Red):** Commits to strategy $\pi_L \in \Pi_L$. A deployment vector $\theta_L \in \mathbb{R}^d$ controlling continuous (x, y) coordinates of turrets, mines, sensors.
- **Follower (Blue):** Observes commitment imperfectly, executes $\pi_F(a|s)$ to maximize J_F.

Follower's Objective (Best Response):

$$\pi_F^*(\pi_L) = \arg\max_{\pi_F} \mathbb{E}_{\tau \sim (\pi_L, \pi_F)} \left[\sum_t \gamma^t r_{F,t} \right] \tag{1}$$

Leader's Objective:

$$\pi_L^* = \arg\max_{\pi_L} J_L(\pi_L, \pi_F^*(\pi_L)) \tag{2}$$

Note: Standard RL fails here because π_L is hidden during execution.

Innovation 1: The Latent Probe Wrapper

From Explicit Queries to Implicit Probing

- Replaces exhaustive explicit querying with implicit behavioral scouting.
- **Probing Phase:** Before the main engagement, Blue executes a short trajectory of length T (e.g., $T = 12$ steps).
- Agent uses a risk-averse exploration baseline driven by a zeroed-out dummy latent vector ($z = \mathbf{0}$).
- The wrapper records the consequence sequence:

$$\tau_{\text{probe}} = \{(s_0, a_0, r_0, s_1), \ldots, (s_T, a_T, r_T, s_{T+1})\} \tag{3}$$

- Encodes damage or proximity warnings without needing a ground-truth map of the enemy deployment.

Innovation 2: Transformer Context Encoder

- Raw trajectories are high-dimensional. We use a Transformer to extract a compact representation of the enemy strategy.
- Transition tuples (s, a, r) are projected to d_{model} and augmented with learnable positional encodings to preserve temporal causality.

Multi-Head Self-Attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

- Focuses on critical moments (e.g., the exact tick a mine was triggered) while ignoring empty transit steps.
- Output is mean-pooled and projected to a low-dimensional latent embedding $z \in \mathbb{R}^{32}$.
- z represents the "concept" of the adversary strategy (e.g., "Kill Zone").

Innovation 3: Hypernetwork Policy

Dynamic Control Logic vs. Simple Concatenation

- Instead of concatenating z to states, z feeds into a secondary generator network g_ϕ.
- The generator outputs the *weights* θ of the primary maneuver policy:

$$\theta_{\text{policy}} = g_\phi(z) \tag{5}$$

$$a_t \sim \pi(s_t; \theta_{\text{policy}}) \tag{6}$$

- **Advantage:** Allows "hot-swapping" entire control logic. Instantly shifts from cautious scouting to aggressive maneuvering.
- **Catastrophic Forgetting:** Mathematically isolates updates. Gradient updates are restricted to the specific operational mode triggered by z, protecting previously learned adversary profiles in the PSRO curriculum.

The PSRO Training Curriculum

Iterative Auto-Curriculum Generation

Initializes a population of Leader strategies Π_L with random policies.

1. **Meta-Follower Optimization:** The Hypernetwork Blue agent is trained via PPO to maximize reward against a uniform distribution of the current Π_L.
2. **Oracle Step (New Leader):** Freeze the Meta-Follower. Train a new Leader policy π_L^{new} to exploit the current Follower (finding the "worst-case" scenario).
3. **Population Expansion:** Add π_L^{new} to Π_L.

Forces the Meta-Follower to learn a generalized mapping from probe trajectories to robust counter-strategies.

Tactical Maneuver Environment

Custom High-Fidelity Simulator (`Gymnasium` interface)

- **Continuous Space:** 40×40 grid representing a contested sector.
- **Entities:**
  - **Blue:** Platoon of $N = 3$ autonomous tanks (continuous differential drive, fire trigger).
  - **Red:** Static defensive network of $K = 3$ heavy turrets, $M = 2$ proximity mines.
- **Terrain Interaction:** Procedurally generated Perlin noise. Roads (1.5x speed), Open (1.0x), Mud/Veg (0.4x).
- **Combat Mechanics:**
  - Turrets use distance-decay probabilistic ballistics: $P(\text{hit}) \propto \max(0, 1 - d/r_{\max})$.
  - Line of Sight (LOS) ray-casting; terrain features occlude vision.
- **Observation Space:** Blue agents have local observability (global state hidden).

Reward Structure

Blue (Follower) Reward:

$$R_{\text{Blue},t} = R_{\text{goal}} + R_{\text{kill}} - R_{\text{damage}} + R_{\text{step}} + R_{\text{approach}}$$

- Goal (+10.0), Turret Kill (+1.5), Damage (-2.0 max/death), Time Penalty (-0.05/step).

Red (Leader) Counter-Objective:

$$R_{\text{Red}} = -\sum_{t} R_{\text{Blue},t} + 3.0 \times N_{\text{killed}} + 5.0 \times I(\text{Blue Failed})$$

- Incentivizes Leader to construct defenses that prevent goal capture AND inflict maximum attrition.

Baselines for Comparison

1. **Original Stackelberg (Random Pre-training):**
   - Standard approach for continuous domains without explicit queries.
   - Domain Randomization followed by frozen weights. Lacks specific real-time adaptation.
2. **Simultaneous MAC-RL (Naïve PPO):**
   - Both agents update simultaneously.
   - Lacks Leader-Follower hierarchy and probing phase. Prone to cycling / non-stationarity.
3. **Concat Baseline (Ablation Study):**
   - Uses the exact same Latent Probe Wrapper and Transformer Encoder.
   - BUT concatenates z to observations ($[obs, z]$) into a standard MLP, rather than using a Hypernetwork.

Experiment A: Robustness Analysis

Learning to Survive Ambushes

- Evaluated against two distinct strategies: "Kill-Zone" (center chokepoint) and "Flanking" (edges).
- Initially, agent survival is $< 10\%$.
- As PSRO population expands, the Hypernetwork identifies the specific threat during the probing phase.
- Successfully toggles behavioral modes (center-rush vs. perimeter-flanking).
- Confirms mitigation of catastrophic forgetting.

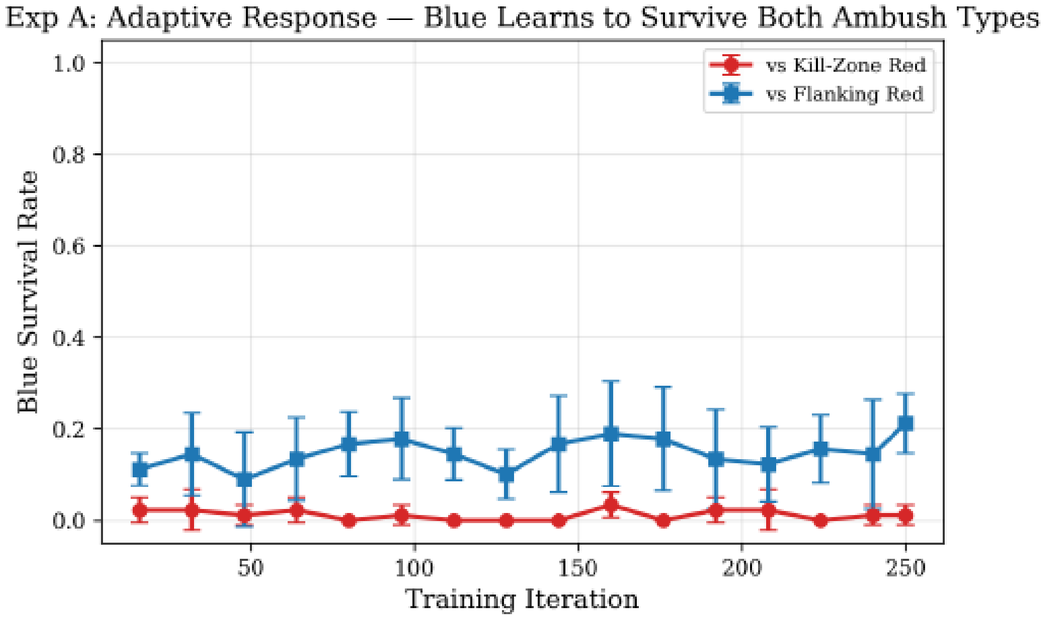


Figure: Agent's survival rate against Kill-Zone and Flanking adversaries improves simultaneously over iterations.

Experiment B: Ablation Study Results

Comparative Performance (Averaged over 5 seeds, 12 eval episodes)

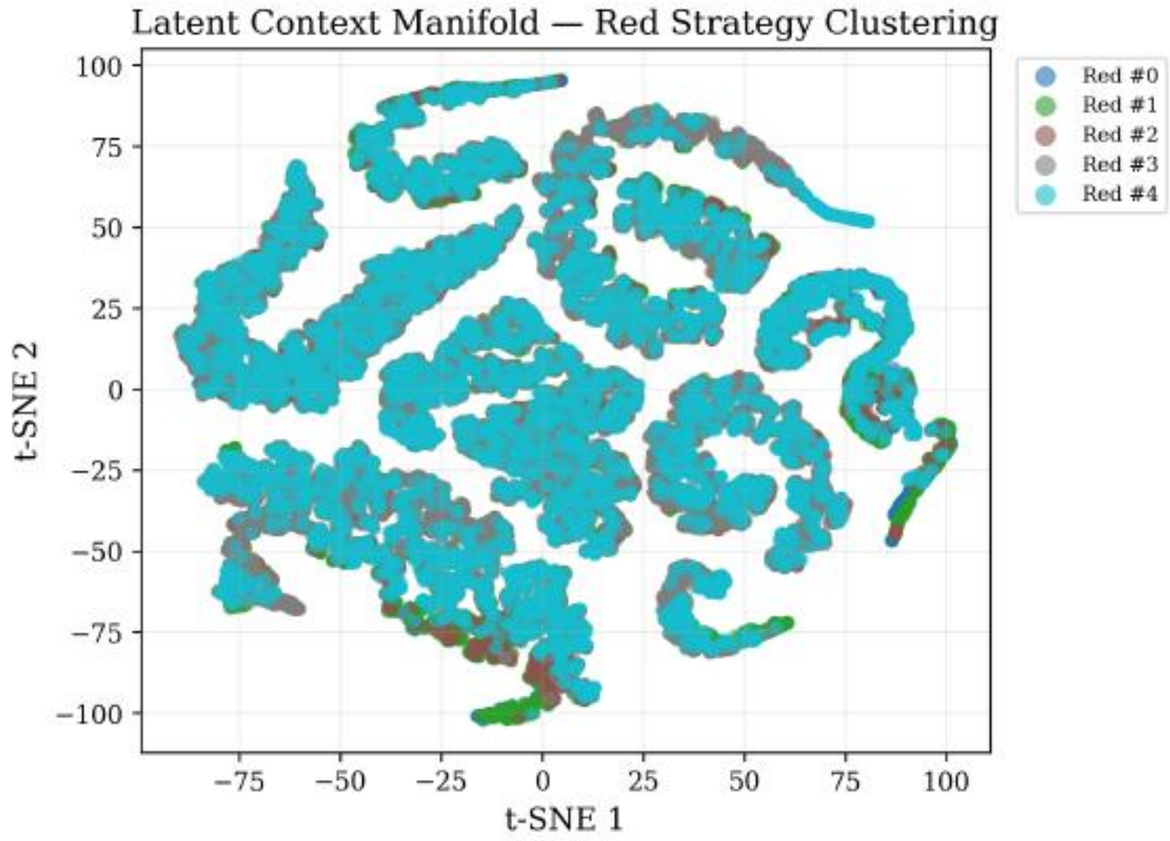
Method	Mean Reward	Mean Survival	Worst-Case Surv.	Unseen Surv.
Hypernet-PSRO (Ours)	-5.36 ± 0.80	0.17 ± 0.02	0.00 ± 0.00	0.28
Concat Baseline	-5.85 ± 0.57	0.16 ± 0.03	0.01 ± 0.02	0.26
Naïve PPO	-5.58 ± 0.53	0.18 ± 0.04	0.01 ± 0.02	0.28

- **Statistical Significance:** Paired t-tests show Hypernet-PSRO significantly outperforms Concat ($p = 0.0149$) and Naïve PPO ($p = 0.0223$).
- Higher negative rewards indicate more efficient navigation and optimized damage mitigation, even when raw binary survival rates appear comparable.

Latent Context Manifold Visualization

t-SNE Projection of $z \in \mathbb{R}^{32}$

- Distinct clustering by Red strategy index.
- E.g., Densely packed Kill-Zone vs. Distributed Area-Denial.
- Spatial distance in the manifold maps to tactical differences.
- Proves the Transformer extracts distinct tactical fingerprints purely from implicit probing (without explicit mapping).



Qualitative Analysis: Adversarial Shaping

Adversarial Digital Twin Behavior

- PSRO "Leader" evolves to concentrate firepower at center map chokepoints and along optimal paths to the objective.
- Emergent optimization: actively systematically denying optimal paths to the Follower.
- Validates the PSRO auto-curriculum as a highly viable Automated Threat Representation (ATR) mechanism.

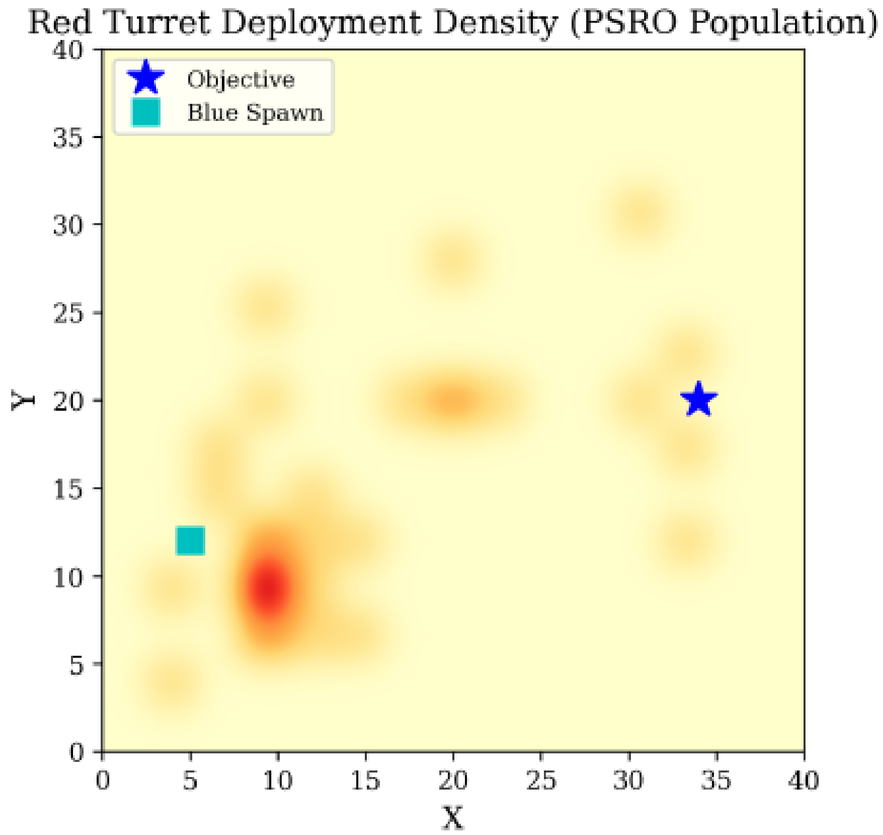


Figure: Heatmap of Red turret deployments.

Conclusion

Summary of Contributions

- Scalable algorithmic framework for computing Stackelberg equilibria in continuous military domains.
- **Latent-Context Hypernetwork Stackelberg PSRO** functions as an *Adversarial Digital Twin*.
- Designed to rigorously stress-test the Army's Robotics Technology Kernel (RTK) prior to physical fielding.
- Implicit probing + Transformer overcomes the explicit state enumeration bottleneck.
- Hypernetworks provide superior inductive bias for rapid behavioral adaptation compared to standard MARL techniques.

Future Work

- **High-Fidelity Simulation Integration:**
  - Transition from 2D tactical simulation to 3D engines (Unreal Engine, Unity) via Ray/RLLib.
  - Validate latent probing under realistic sensor noise and 3D terrain physics.
- **Dynamic Red Team Agents:**
  - Extend from static deployments to kinetic, reactive threats (e.g., mobile enemy armor, loitering munitions).
- **Transition to DoD Engineering:**
  - Integrate frameworks with ROS-M (Robot Operating System - Military).
  - Bridge the gap between simulation-based T&E and real-world operational readiness via MOSA compliance.

Acknowledgments & Questions

Thank You

Acknowledgment: The authors acknowledge the technical and financial support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-24-2-0001 U.S. Army DEVCOM Ground Vehicle Systems Center (GVSC) Warren, MI.

Questions?