Mixed Formal-Methods Pretraining for Safe MARL: Implicit affordance aware planning via structured pretraining priors

Research Summary (2026) Safety · MARL · Formal Methods

1 Short Summary

Consider a drone in a cluttered, dynamic environment—other agents with unknown intent, obstacles emerging from occlusion, corridors feasible only within a narrow time window. An expert pilot, from a few seconds of partial observations, builds an implicit forecast of where everything will be, identifies a gap about to open or close, and commits. They are not reacting to the environment. They are *reading* it—and finding maneuvers a reactive controller would never attempt.

MARL can learn effective collision avoidance and coordination given enough training, and recent methods have made meaningful progress on sample efficiency and safety-aware exploration. But there is a gap between "learns to avoid collisions" and "learns to exploit the feasibility structure of a dynamic scene." The latter requires capabilities that are hard to acquire from reward signal alone: recognizing that two converging agents will create a brief gap, that decelerating now opens a merge window later, or that a coordinated yield unlocks a corridor invisible under greedy control. These are not failures of RL — they are consequences of learning feasibility from scratch in high-dimensional, partially observable, multi-agent settings, where the relevant structure is difficult to discover through exploration.

Formal methods—HJ reachability, Tube-MPC, CBFs—can characterize these windows exactly: the reachable set, the feasible tube, the time-to-boundary. But they are almost always applied as runtime shields. The policy never absorbs the *structure* of feasibility.

The idea. Use formal-methods solutions as *training data*, not runtime constraints. Pre-train a policy on a large corpus of offline reachability solutions, trajectory tubes, and safety certificates so that it internalizes feasibility structure under dynamics and interaction. The goal is a policy with *emergent implicit planning*: one that reads partial observations, predicts how envelopes evolve, and commits to dynamic maneuvers with the confidence of an agent that *knows* the envelope.

What (in theory) this buys you beyond standard safety:

- **Implicit prediction from partial observability.** The policy learns to infer future reachable sets from a short observation window—closing distances, relative velocities, trajectory fragments—without an explicit prediction module.
- **Edge-case exploitation, not just avoidance.** Instead of steering around clutter, the policy finds and commits to tight feasible corridors that exist only briefly.
- **Multi-agent strategic reasoning.** Coordination emerges because the policy understands how its actions reshape the joint reachable space—yielding, accelerating, or repositioning to open windows for itself and others.
- **Dramatically faster useful training.** Early checkpoints are already competent enough to test, because the policy does not need thousands of crashes to discover what "feasible" means.

2 Why Mix Methods?

Our instinct is that no single formal method covers enough of the safety landscape to serve as a universal pre-training source. Each one is good at something different:

Method (examples)	**What it gives you**	**Where it breaks down**
HJ Reachability	Exact unsafe sets, value functions	Exponential in state dimension
Tube-MPC	Robust trajectory corridors	Conservative, local
CBF certificates	Pointwise safety margins	Misses multi-step interactions
Self-play logs	Naturalistic multi-agent behavior	No formal guarantees

The intuition is analogous to multimodal foundation models: vision and language alone each miss structure that appears when you train on both. Here the "modalities" are formal-methods outputs at different fidelity–scalability tradeoffs. A mixed corpus could let the network inherit *precision* from formal methods where they are tractable and *coverage* from simulation where they are not.

We are not certain this is true. It is possible that one method (e.g., HJ reachability alone, at sufficient scale) is enough, or that mixing introduces more confusion than benefit due to conflicting supervision signals. The point of this proposed project is to find out (ablations)

3 What We Think Might Be True (And What It Would Take to Show It)

There are three core intuitions. None of them are established results—they are bets that guide the research. For each one, here is the intuition, why we might be wrong, and what evidence would settle it.

Intuition 1: Pre-training on safety data helps early training

The instinct. A policy pre-trained on formal-methods solutions should violate safety constraints less during the early phase of online RL, compared to a policy trained from scratch. The pre-trained network has already seen what unsafe regions look like and should avoid them before the RL reward signal has had time to teach the same lesson.

Why we might be wrong. The pre-training distribution could be too narrow or too different from the online environment, so the prior washes out immediately. Or a well-shaped reward penalty might achieve the same effect more cheaply.

What would prove it. Compare learning curves (violations per episode vs. training step) for: (a) pure MARL from scratch, (b) MARL with dense safety-penalty reward shaping, (c) pre-trained then fine-tuned. If (c) shows a large reduction in early violations—ideally order-of-magnitude—and the gap persists for a meaningful portion of training, the intuition holds.

Intuition 2: Mixing methods beats any single source

The instinct. A corpus mixing HJ reachability, Tube-MPC, and self-play trajectories produces a better policy prior than any single source alone, because they cover complementary failure modes. HJ gives precise low-dimensional safety sets; self-play gives high-dimensional multi-agent coordination patterns; Tube-MPC fills in trajectory-level robustness.

Why we might be wrong. Conflicting labels are a real problem: reachability may call a state unsafe that self-play trajectories pass through routinely. The network might learn to ignore the formal signals in favor of the more abundant empirical ones, or vice versa. It is also possible that "more data from one good source" beats "less data from many sources" at matched compute.

What would prove it. Single-source baselines at *matched data budget*: HJ-only, Tube-MPC-only, self-play-only, each with the same total number of training examples. If the mixed model outperforms all single-source models on both safety and task metrics, the mixing is doing real work. Mixing-ratio sweeps and per-source ablations would further isolate what each modality contributes.

Intuition 3: The pretrained network learns safety structure, not just imitation

The instinct. The transformer does not just memorize safe trajectories—it learns something deeper, like an implicit value function over the safe set (analogous to the BRT value function from HJ reachability). If true, the pretrained representations should predict time-to-boundary, safe-set membership, and feasibility in scenarios never seen during pre-training.

Why we might be wrong. The network might simply memorize a set of heuristics that happen to correlate with safety in-distribution but fail under shift. Or the representations might be useful but not interpretable as value-function-like objects.

What would prove it. Test OOD generalization: does the pretrained model maintain safety in environments with novel agent counts, obstacle densities, or dynamics parameters unseen during pre-training?

4 Method (Briefly)

Simulation. Procedurally generated N-agent environments in Isaac Sim with randomized dynamics, time-varying constraints, and varied task objectives (navigation, formation, coverage).

Data. For each scenario, compute offline safety labels from two or more formal methods. Start with the two most complementary sources—HJ reachability (precise, low-dim) and self-play near-misses (scalable, high-dim)—and add Tube-MPC and CBFs incrementally.

Pre-train a transformer backbone on the mixed corpus (safe-set value regression, feasibility classification, safe-action imitation).

Fine-tune with multi-agent RL under task reward. Compare unconstrained reward, penalty-based shaping

Hard-case mining. Self-play finds near-misses → formal planner solves them → augment corpus → retrain. Keeps the data relevant as the policy improves.

5 Risks to Watch

Signal conflict. Different methods may disagree on whether a state is safe. This is not just a data-cleaning problem—it is a core design decision. We need to decide: hierarchy (formal overrides empirical), uncertainty weighting, or let the network arbitrate. This choice may end up being the main contribution.

Scope. Implementing HJ + Tube-MPC + CBF + self-play is a lot of infrastructure. Maybe start with two sources and expand only if the two-source version shows clear promise.

Forgetting. Fine-tuning could erase the safety structure learned during pre-training. Mitigations: KL regularization toward pretrained weights, early stopping on safety metrics, and continuous hard-case mining to reinforce safety-critical behavior.