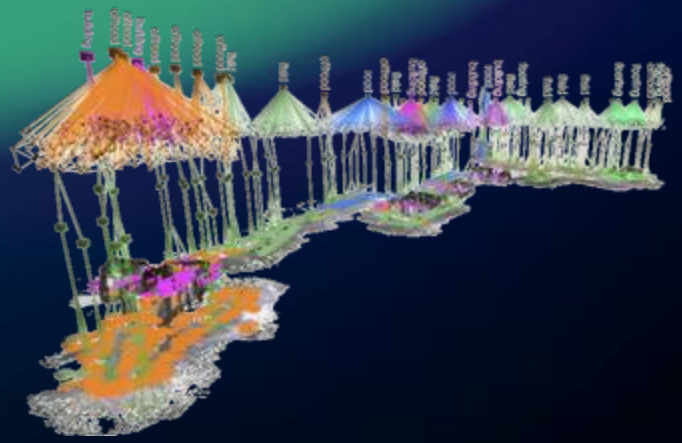
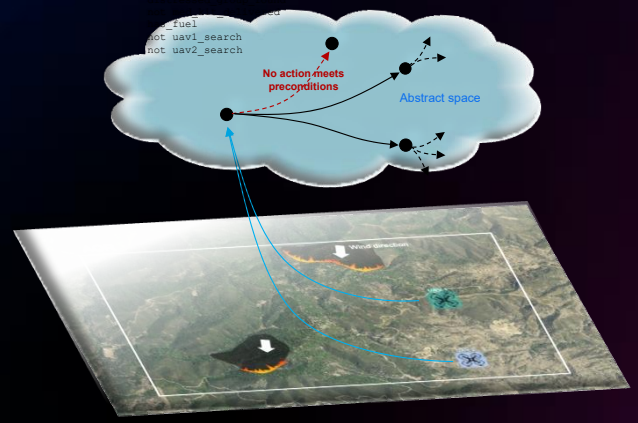
Integrating Neuro-Symbolic Approaches in Autonomous Systems





Alberto Speranzon

Advanced Technology Laboratories (ATL)
Lockheed Martin

AVIATE Seminar – 8/8/2025

LOCKHEED MARTIN

Acknowledgements

- **LM:**
  - Tom Lee
  - Kevin Bohlin
- **MIT:**
  - Luca Carlone
  - Nathan Hughes
  - Jared Strader
- **ASU:**
  - Rushan Karia
  - Siddharth Srivastava
  - Pulkit Verma

- J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, “Indoor and Outdoor 3D Scene Graph Generation Via Language-Enabled Spatial Ontologies”, IEEE Robotics & Automation Letter, 2024.
- R. Karia, P. Verma, A. Speranzon, S. Srivastava, “Epistemic Exploration for Generalizable Planning and Learning in Non-Stationary Settings”, ICAPS 2024.
- A. Speranzon et al. “Challenge Problems in Developing a NS OODA Loop”, Workshop on Neuro-Symbolic Learning and Reasoning, 2023

Assured and Trusted Collaborative Autonomy

- Human-autonomy collaborative teams
- Multi-scale and multi-domain
- Scalability and compositionality
- Assurance
- Uncertainty and ambiguity management

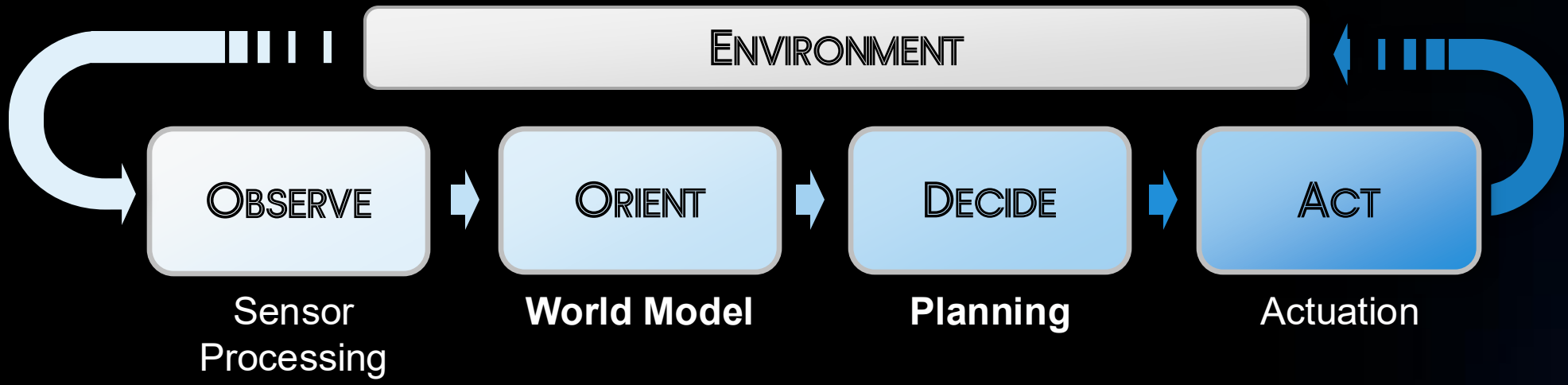
LOCKHEED MARTIN

Assured and Trusted Collaborative Autonomy

- Human-autonomy collaborative teams
- *Multi-scale and multi-domain*
- *Scalability and compositionality*
- *Assurance*
- Uncertainty and ambiguity management

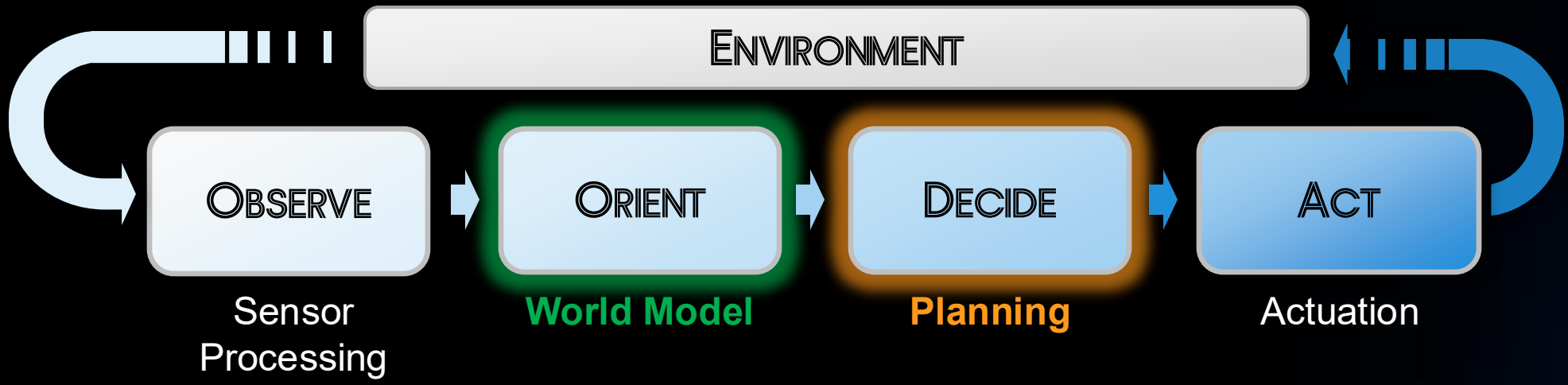
LOCKHEED MARTIN

Why NS in the OODA Loop?



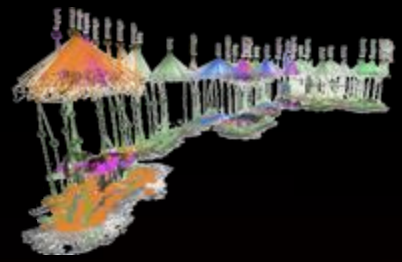
- Symbolic combined with subsymbolic abstractions:
  - Provide mechanisms to *incorporate symbolic priors* (e.g., from field manuals), a key aspects in applications domains where data is not readily available
  - In general, lead to more *"compressed" state/action spaces*
  - More *interpretable by humans* and more *suitable for formal analysis*, a critical need in safety critical applications
  - Leverages advances in *statistical machine learning*
  - Exploit *compositional properties* of symbolic abstractions

NS Methods in the Orient and Decide Blocks



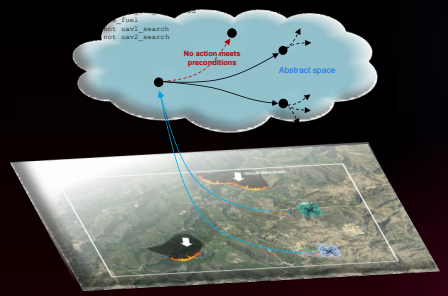
World Model:

- Hierarchical NS representations



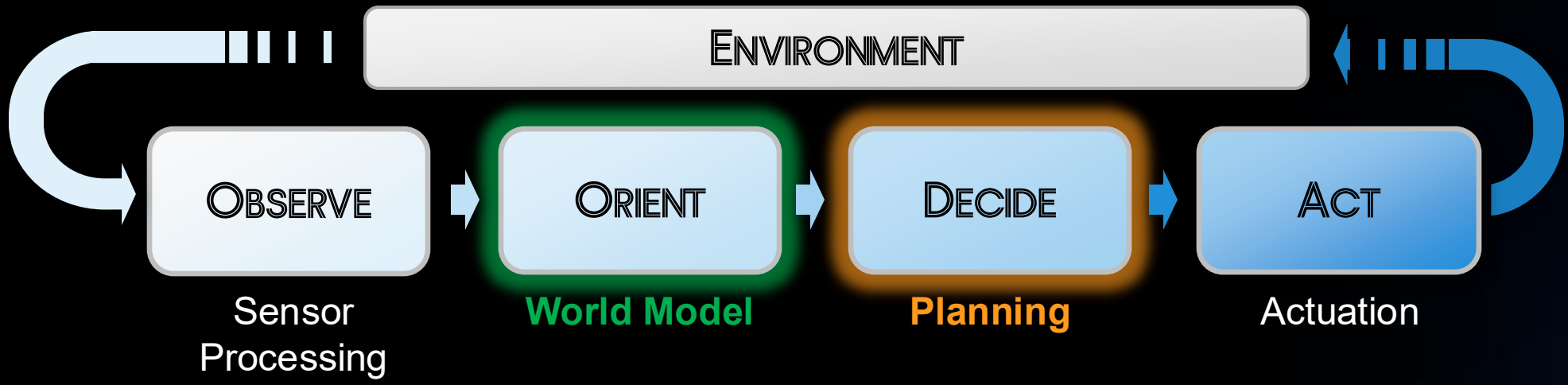
Planning:

- Relational abstractions for scalable planning



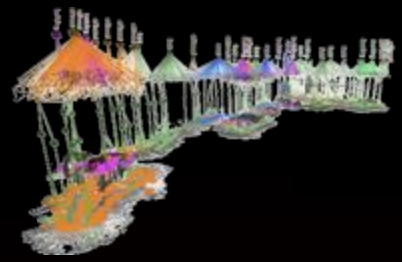
LOCKHEED MARTIN

NS Methods in the Orient and Decide Blocks



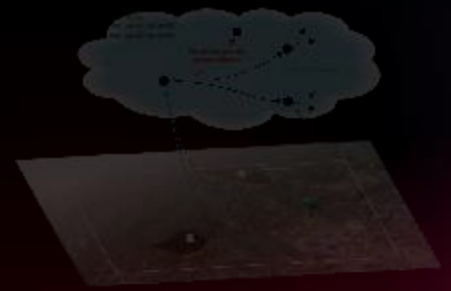
World Model:

- Hierarchical NS representations



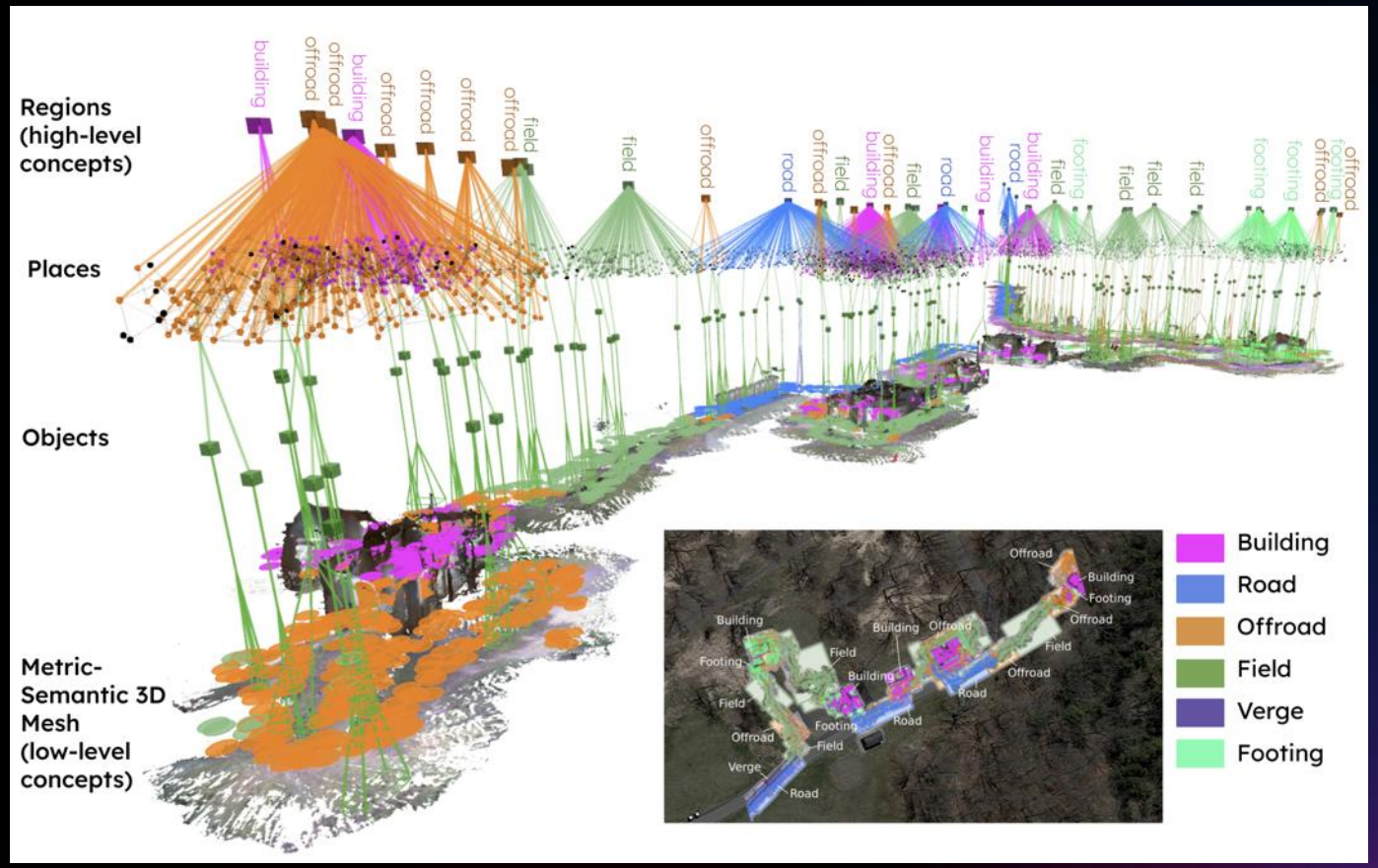
Planning:

- Relational abstractions for scalable planning



Dynamics Scene Graphs as NS World Models

- Dynamic Scene Graphs (DSGs) are a *hierarchical 3D representation* of complex environments
- Nodes and edges represent spatial *concepts* and their *relations* at different level of "granularity"
- Nodes and edges are grouped into layers at *different level of abstraction*



Layers in the hierarchy:

- L1 (bottom): metric-semantic 3D mesh
- L2: objects and (dynamic) agents
- L3: places (a topological representation of free space)
- L4: rooms or subregions
- L5: buildings or regions

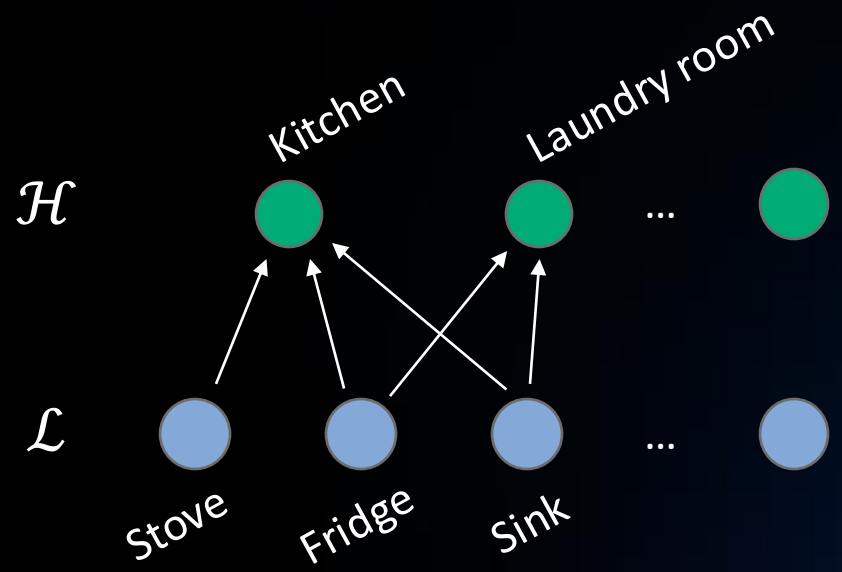
J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and Outdoor 3D Scene Graph Generation Via Language-Enabled Spatial Ontologies", IEEE Robotics & Automation Letter, 2024.

Spatial Ontology

- A DSG is learned using various deep & graph neural networks (DNNs & GNNs) working at *different level of abstractions*
- As the concepts and relations refer to spatially distributed objects and regions, it would be beneficial to constraint DNNs/GNNs to satisfy relations captured by a *spatial ontology*

Spatial Ontology

- A DSG is learned using various deep & graph neural networks (DNNs & GNNs) working at *different level of abstractions*
- As the concepts and relations refer to spatially distributed objects and regions, it would be beneficial to constraint DNNs/GNNs to satisfy relations captured by a *spatial ontology*
- A *spatial ontology* is a bipartite graph where:
  - Nodes are either *low-level concepts*, $\mathcal{L}$, or *high-level concepts*, $\mathcal{H}$
  - Edges represent *inclusion relations*, $r(\ell, h)$, between low-level concepts and high-level concepts
- Although we consider only a two-level hierarchy one can extend to multiple levels



$$r : \mathcal{L} \times \mathcal{H} \to \Sigma$$

where

$$r(\ell, h) \in \{0,1\}$$

or

$$r(\ell, h) \in [0,1]$$

Spatial Ontology Generation from LLM

- Generate a spatial ontology by querying a LLM

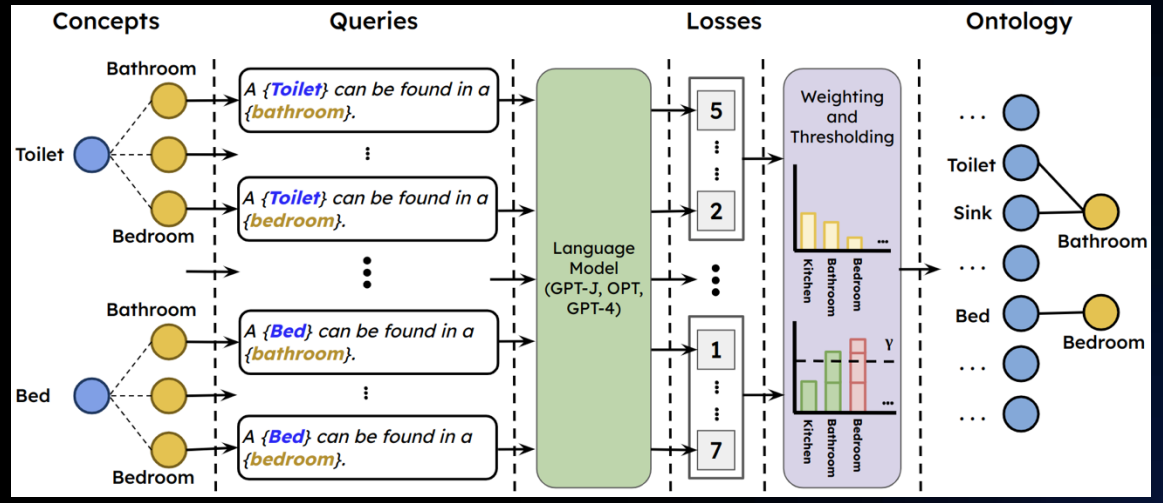
Two approaches

- *Text Scoring*:
  - Assume there is an edge (ℓ, h) for every pair of low- and high-level concepts
  - Generate a weight (and threshold)

$$w(\ell_i, h_j) = \text{softmax}\left(\Lambda\left(g_{score}(\ell_i, h_j)\right)\right)$$

$$g_{score}(\ell_i, h_j) = \text{“}\ell_i \text{ is often found in } h_j\text{”}$$

$$\Lambda(W) \approx \log p(W)$$



LOCKHEED MARTIN

Spatial Ontology Generation from LLM

- Generate a spatial ontology by querying a LLM

Two approaches

- *Text Scoring*:
  - Assume there is an edge (ℓ, h) for every pair of low- and high-level concepts
  - Generate a weight (and threshold)

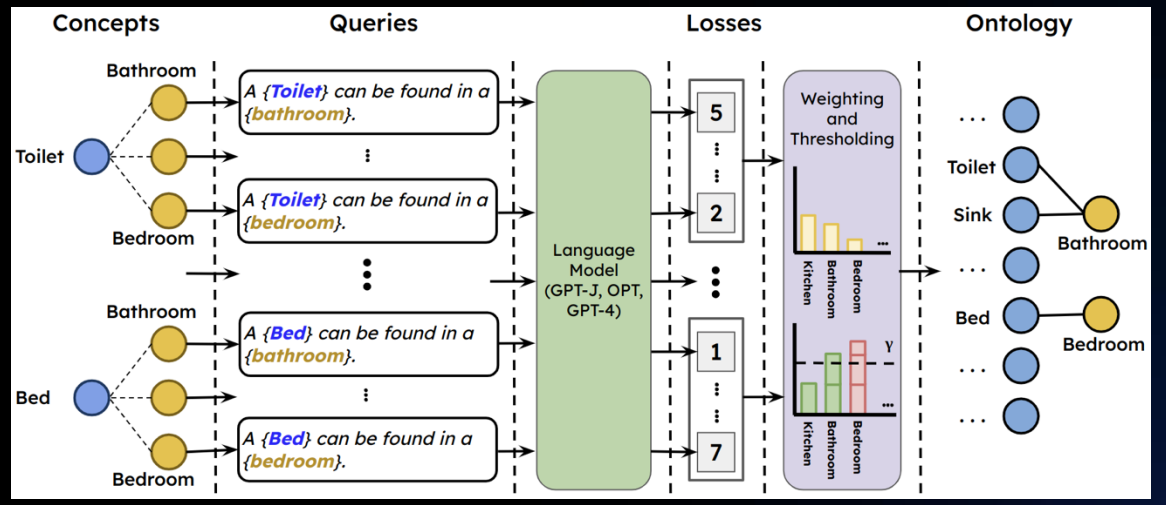
$$w(\ell_i, h_j) = \text{softmax}\left(\Lambda\left(g_{score}(\ell_i, h_j)\right)\right)$$

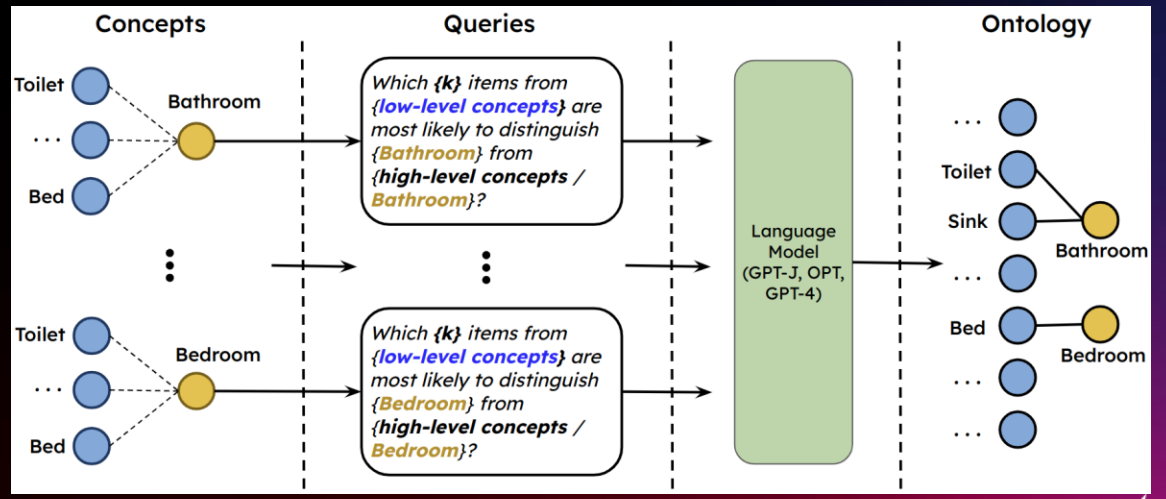
$$g_{score}(\ell_i, h_j) = \text{“}\ell_i \text{ is often found in } h_j\text{”}$$

$$\Lambda(W) \approx \log p(W)$$

- *Text Completion*:
  - Use the score:

$$g_{comp}(\{\ell_i\}, h_j) = \text{“Which } k \text{ items, } \{\ell_i\}\text{, from } \mathcal{L}\text{, are most likely to distinguish } h_j \text{ from } \mathcal{H} \setminus h_j\text{. Answer with a list using exact strings in } \mathcal{L}\text{.”}$$





Constraining Learning via Logic Tensor Networks (LTN)

- LTN is a *model-theoretic approach* to NS AI that "constraints" the learning of a sub-symbolic model via a loss function
- LTN is a *first-order language* whose semantics is defined by a *MV-algebra* (fuzzy logic)
- The interpretation of constants, variables and functions is that of *real-valued tensors*
- Unary and binary connectives as well as predicates *operates on tensors* based on the algebra operations
- Functions and predicates are parametrized by *learnable parameters*

Given a set of formulas $\mathcal{K}$, learning the parameters is posed as

$$\theta^* = \arg\max_{\theta \in \Theta} \operatorname{SatAgg}_{\phi \in \mathcal{K}} I(\phi|\theta) - \lambda \mathcal{J}(\theta)$$

- $\operatorname{SatAgg}: [0,1]^* \to [0,1]$ is an aggregating operator over the formulas in $\mathcal{K}$
- $\mathcal{J}(\theta)$ is any other loss function and/or regularization term

S. Badreddine, A.D.A Garcez, L. Serafini and M. Spranger, "Logic tensor networks", *Artificial Intelligence*, *303*, 2022.

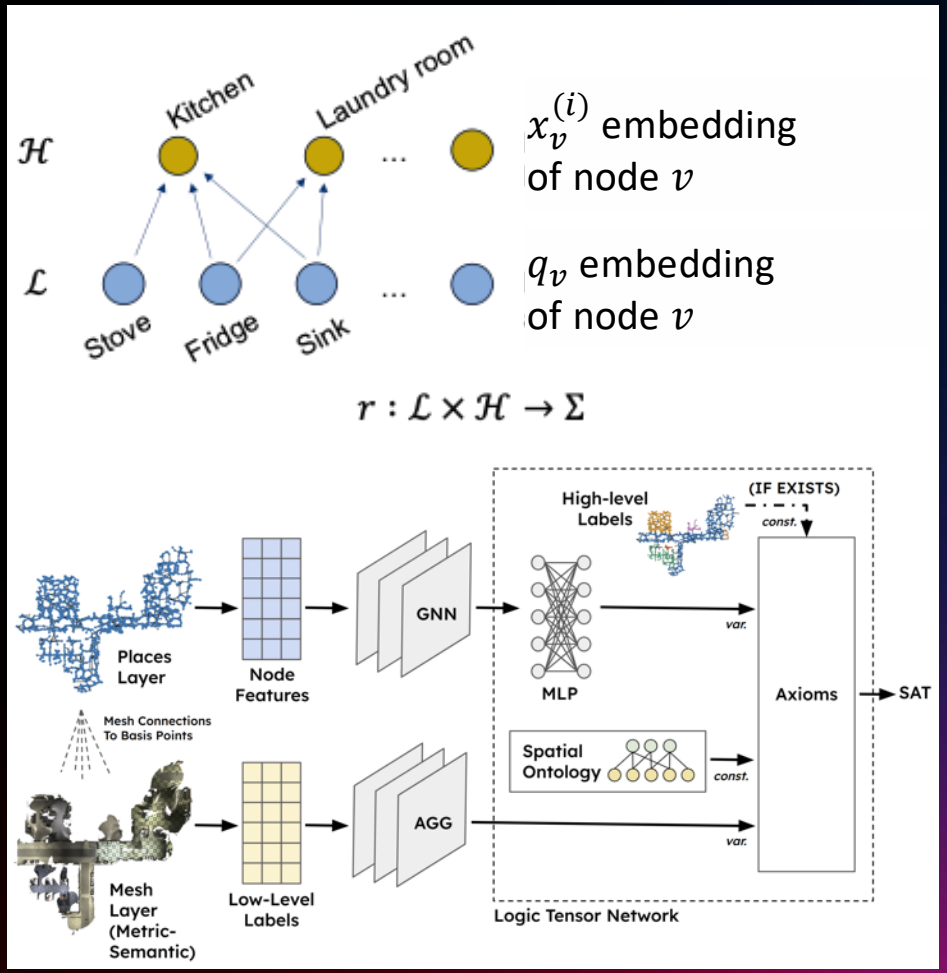
Learning to Build DSGs with LTN

Concepts and relations between concepts in the spatial ontology are made into predicates

Equivalence Predicates:

- Given a label y_i for and the embedding $x_v^{(i)}$ of a (place) node $v \in \mathcal{G}$ then:
- $I(\texttt{IsClassOf}|\theta): (x_v^{(i)}, y_i) \mapsto y_i^T \cdot \mathrm{MLP}_\theta(x_v^{(i)})$

$$\phi_{equiv}: \forall x_v^{(i)}\ \texttt{IsClassOf}(x_v^{(i)}, y_i)$$



Learning to Build DSGs with LTN

Concepts and relations between concepts in the spatial ontology are made into predicates

Equivalence Predicates:

- Given a label y_i for and the embedding $x_v^{(i)}$ of a (place) node $v \in \mathcal{G}$ then:
- $I(\texttt{IsClassOf}|\theta)\colon (x_v^{(i)}, y_i) \mapsto y_i^T \cdot \mathrm{MLP}_\theta(x_v^{(i)})$

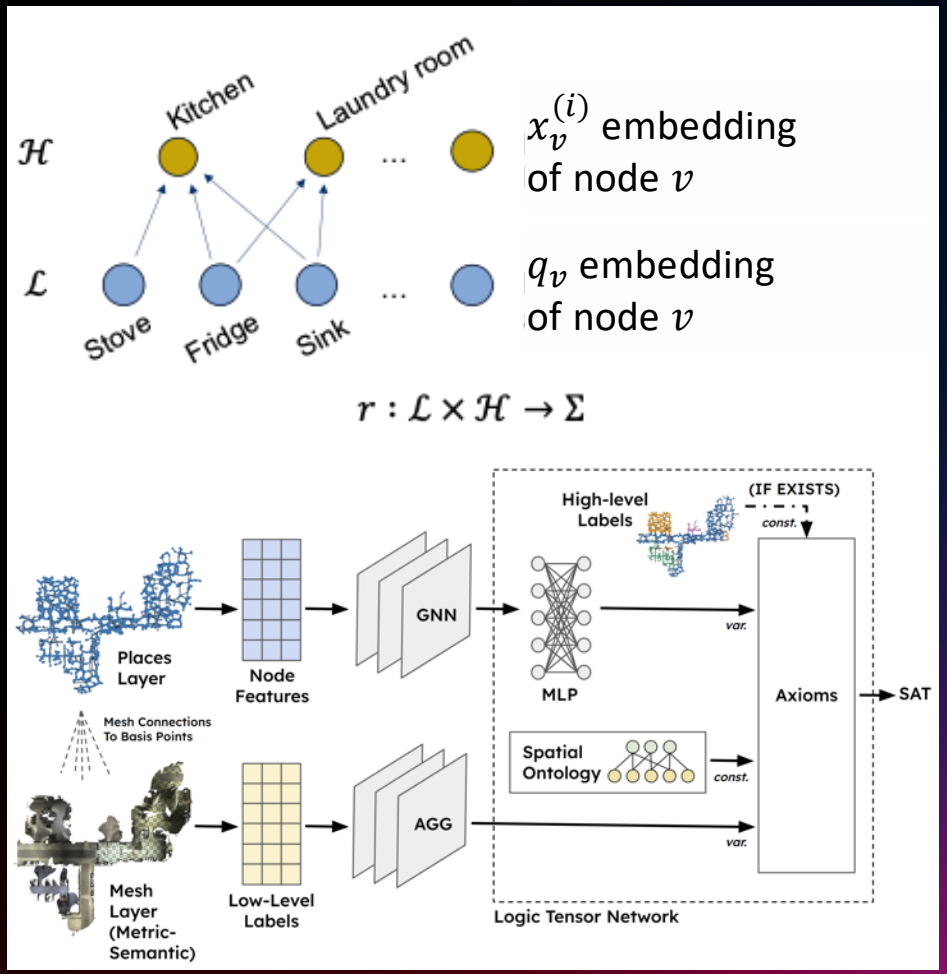
$$\phi_{equiv}\colon \forall x_v^{(i)}\ \texttt{IsClassOf}(x_v^{(i)}, y_i)$$

Inclusion Predicates:

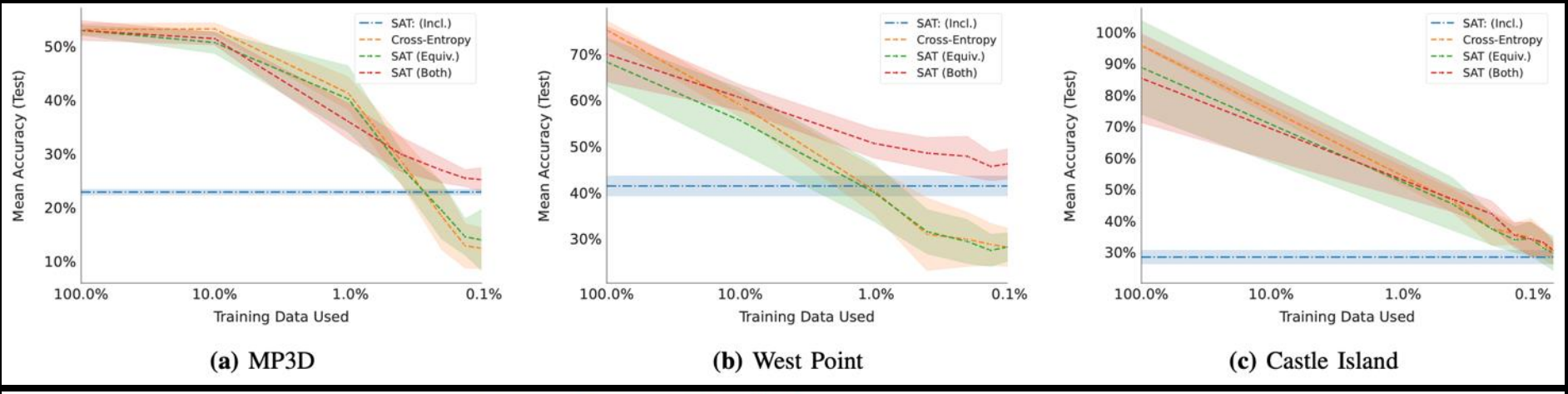
- q_v low-level concepts associated to a (place) node v
- $I(\texttt{IsValid})\colon (r, q_v) \mapsto \mathrm{sum}(r \cdot q_v)$
- $I(\texttt{IsSimilar})\colon (x_v, r, q_v) \mapsto \mathrm{MLP}_\theta\left(x_v^{(i)}\right)^T \cdot r \cdot q_v$

$$\phi_{incl}\colon \forall x_v^{(i)}, q_v\ \texttt{IsValid}(r, q_v) \rightarrow \texttt{IsSimilar}(x_v^{(i)}, r, q_v)$$

- $\mathcal{K} = \{\phi_{equiv}, \phi_{incl}\}$

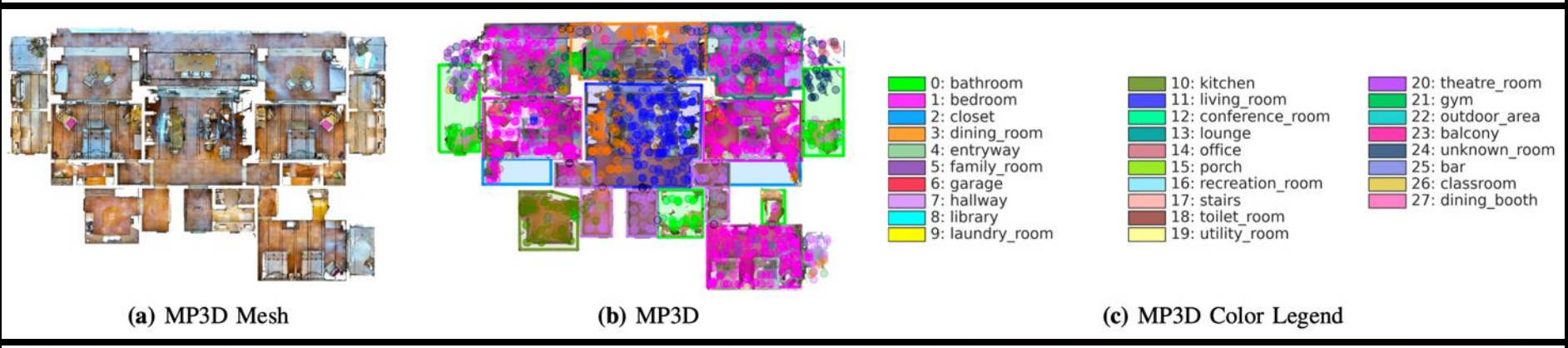


(Some) Results



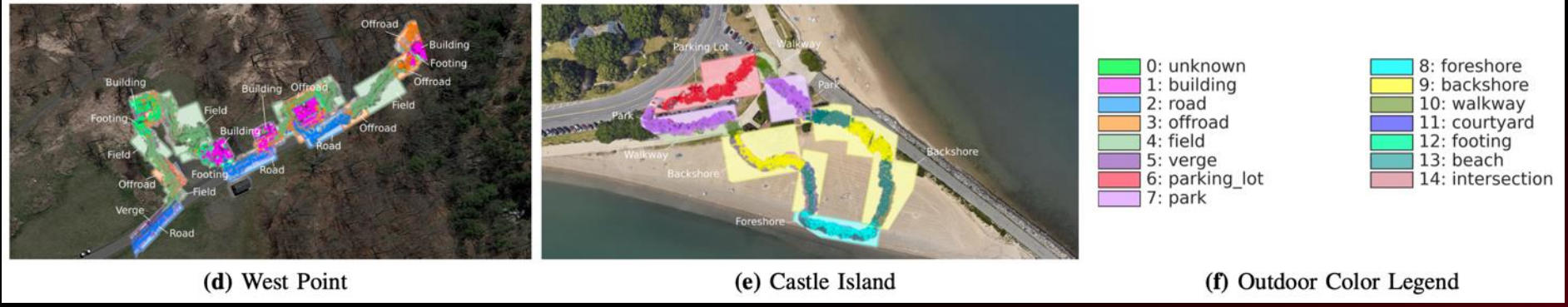
(a) MP3D (b) West Point (c) Castle Island

10-1000x less labeled data yields graceful degradation of performance



(a) MP3D Mesh (b) MP3D (c) MP3D Color Legend

Masked labels: bathroom, kitchen, hallway, bedroom, living room, and family room. No data held out during testing.

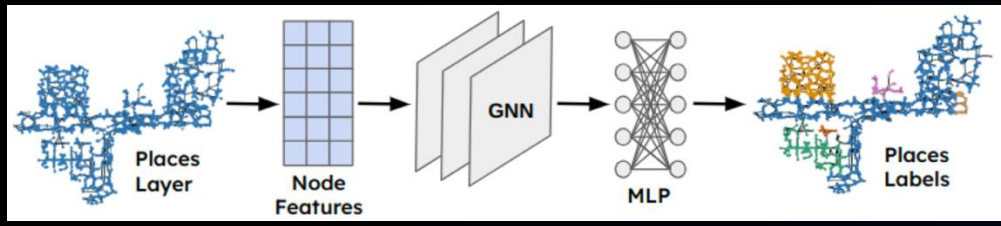


(d) West Point (e) Castle Island (f) Outdoor Color Legend

J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and Outdoor 3D Scene Graph Generation Via Language-Enabled Spatial Ontologies", IEEE Robotics and Automation Letter, 2024.

LOCKHEED MARTIN

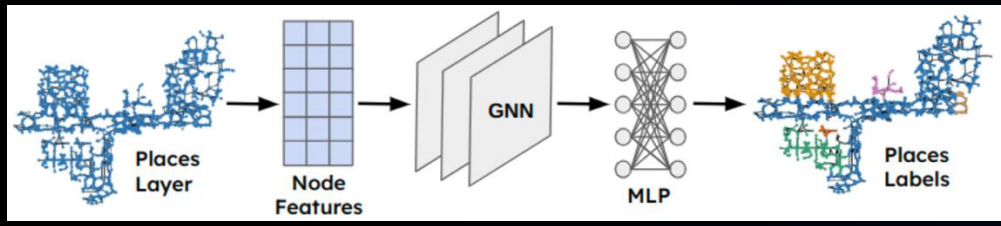
Some Important Challenges



- There are concepts that are “easily” defined via *symbolic relationships* with other concepts, however certain concepts are better defined from *data*

How do we choose? When should we use an ontology and when should we use data?

Some Important Challenges



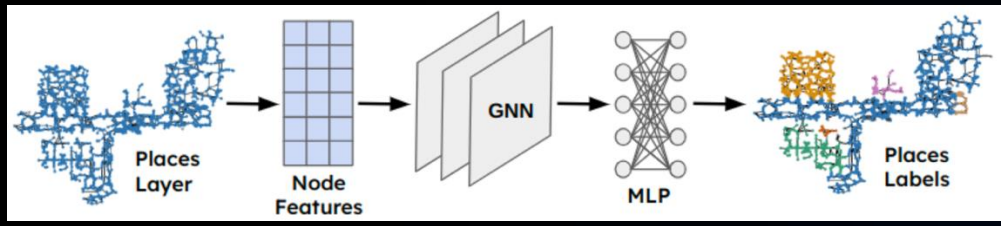
- There are concepts that are “easily” defined via *symbolic relationships* with other concepts, however certain concepts are better defined from *data*

- At inference time the LTN is “removed”, however it seems reasonable that one would like to use the *predicates to monitor* the consistency of the classification based on the ontology

How do we choose? When should we use an ontology and when should we use data?

Should we use this information at runtime? How?

Some Important Challenges



- There are concepts that are “easily” defined via *symbolic relationships* with other concepts, however certain concepts are better defined from *data*

- At inference time the LTN is “removed”, however it seems reasonable that one would like to use the *predicates to monitor* the consistency of the classification based on the ontology

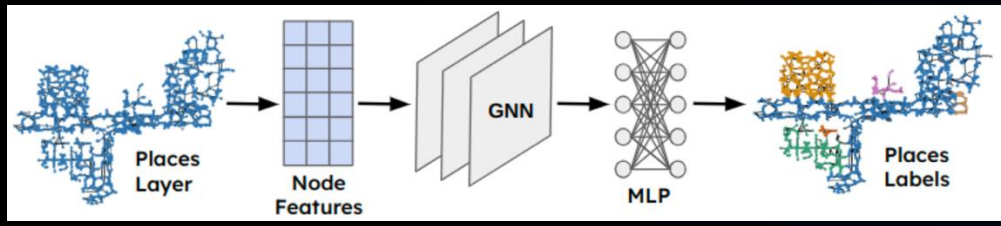
- (Approximate) *reasoning could be executed at runtime* on additional predicates/axioms in the knowledge base

How do we choose? When should we use an ontology and when should we use data?

Should we use this information at runtime? How?

If we have a large ontology, we will likely not want to have too many predicates constraining the network. Can we use others to reason about the generated DSG?

Some Important Challenges



- There are concepts that are “easily” defined via *symbolic relationships* with other concepts, however certain concepts are better defined from *data*

  How do we choose? When should we use an ontology and when should we use data?

- At inference time the LTN is “removed”, however it seems reasonable that one would like to use the *predicates to monitor* the consistency of the classification based on the ontology

  Should we use this information at runtime? How?

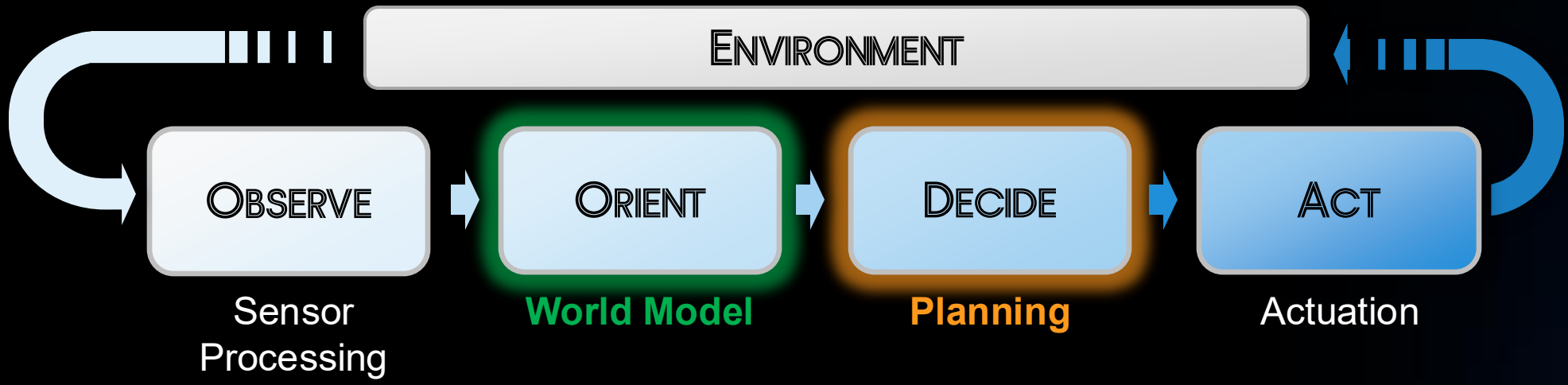
- (Approximate) *reasoning could be executed at runtime* on additional predicates/axioms in the knowledge base

  If we have a large ontology, we will likely not want to have too many predicates constraining the network. Can we use others to reason about the generated DSG?

- The interpretation given to connectives and quantifiers needs to *balance three properties*:
  - They are (valid) operations in the MV-algebra
  - They yield well behaved gradients during backprop
  - The underlying lattice has rich and useful properties for approximate reasoning

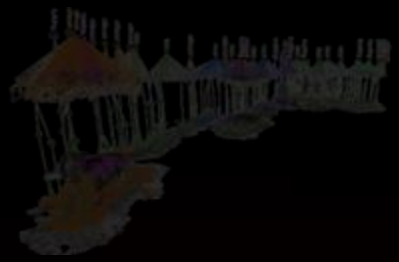
  Currently, items 1. and 2. are really the focus, but 3. should be taken into consideration if we want to do reasoning

NS Methods in the Orient and Decide Blocks



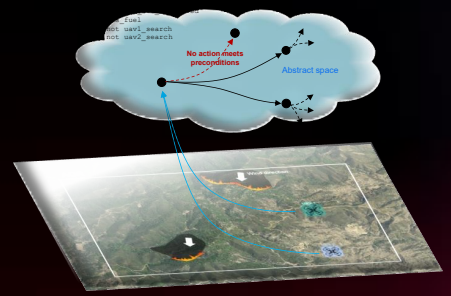
World Model:

- Hierarchical NS representations



Planning:

- Relational abstractions for scalable planning



LOCKHEED MARTIN

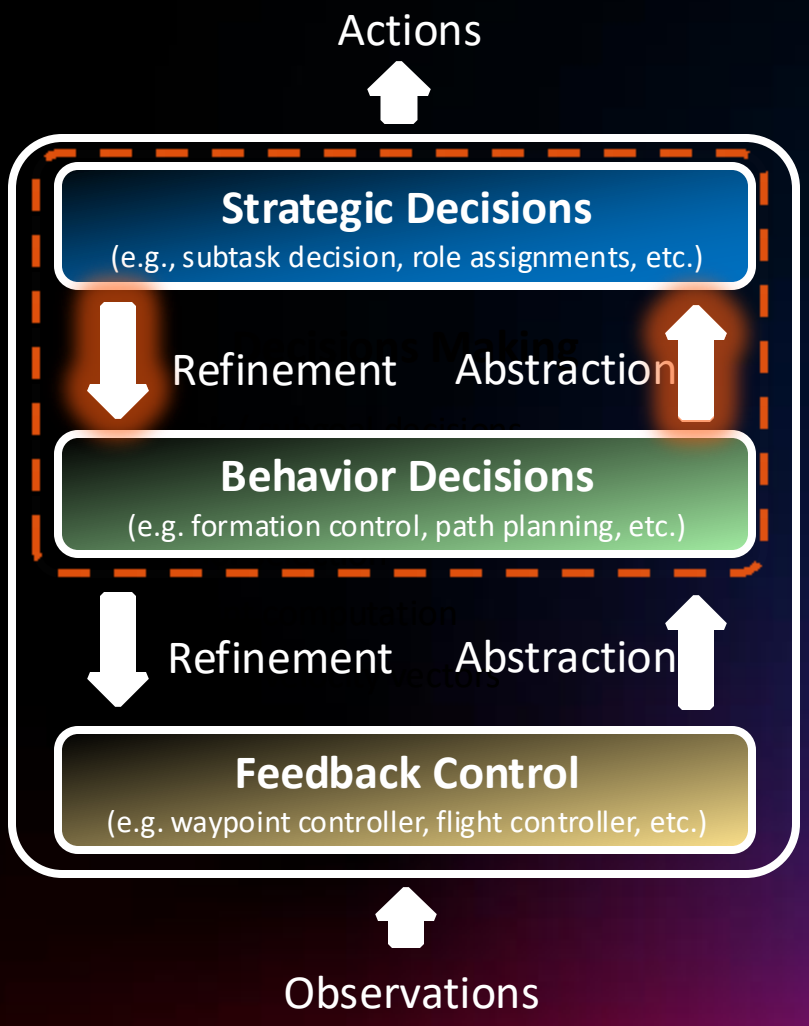
Large Scale Multi-Domain & Multi-Agent Missions

- Deployment of large-scale systems in complex (adversarial) environments require solutions that are *highly scalable within a domain* and *across domains*
- Deployment of complex multi-domain multi-agent autonomous system require *compositional design* paradigms that provide *system assurance* and are guaranteed to **obey** *rescue operations requirements* and *specific mission rules*
- Humans will be involved in the mission, providing some level of supervisory control, and thus we need to develop *interpretable solutions* (at least at the strategic level of decision)

Large Scale Decision Making & Role of Abstraction

- The sample complexity of multi-agent reinforcement learning (MARL) *grows exponentially*, i.e., the size of the action space of grows exponentially with the size of the team[1]
- Teams will be *inherently heterogenous* both within the same domain (e.g., different sensor capabilities) and especially across domains (e.g., different platforms subject to different type of constraints)
- End-to-end solutions may not scale well: heterogeneousness will require re-training / re-tuning policies, simulation of very large multi-agent systems in complex dynamic environment (with adversarial entities) is very time consuming to develop and sample

➢**Approach:** Hierarchical neuro-symbolic abstractions



[1] K. Zhang, S.M. Kakade, T. Basar, L.F. Yang, "Model-Based Multi-Agent RL in Zero-Sum Markov Games with Near-Optimal Sample Complexity", Journal of Machine Learning (24), 2024.

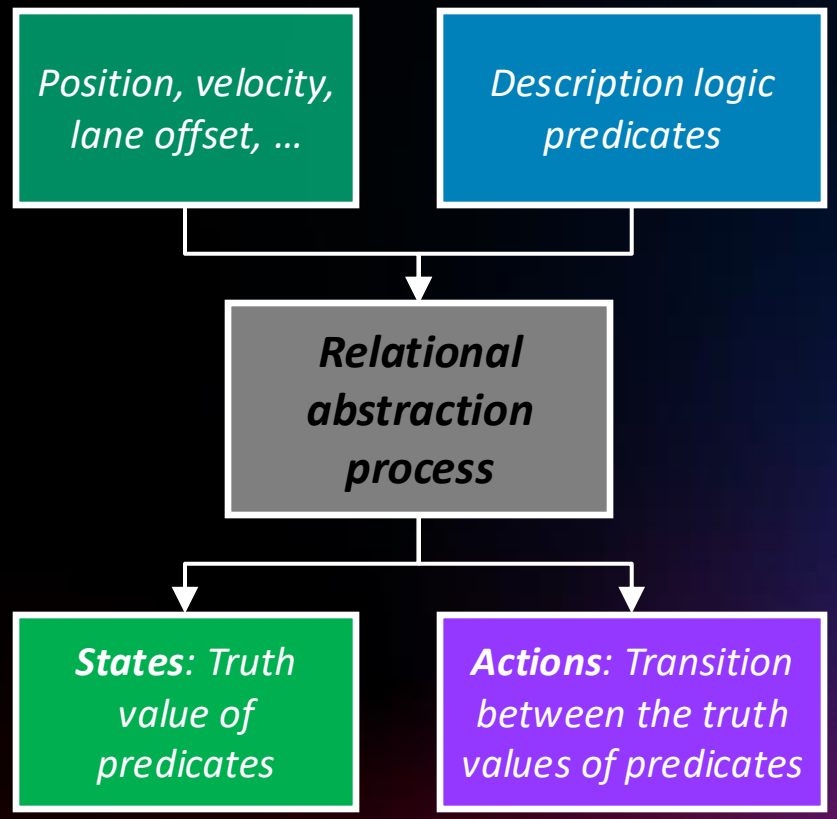
NS Approach to Decision Making

Relational abstractions lift the ground states and actions of a Markov Decision Process to abstract states and actions modeled via *description logic predicates*

We consider a *Relational MDP*

$$M = \left(\langle P^{\uparrow}, A^{\uparrow}\rangle, O, S, A, \delta, \rho, s_0, g\right)$$

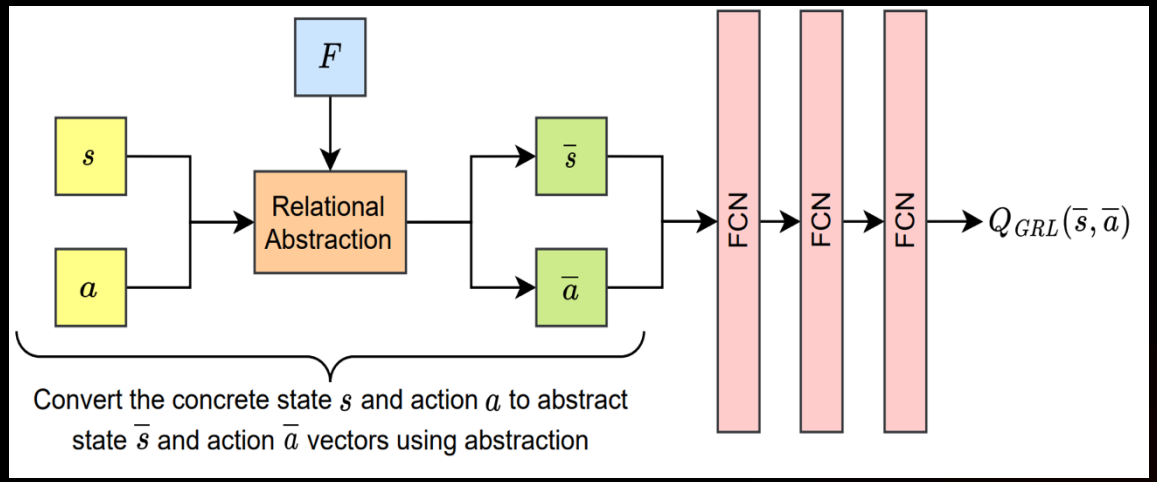
- $P^{\uparrow} = \{\texttt{RobotAt}(p),\ \texttt{ObjectAt}(q), \texttt{ObjectOnRobot}(p, o), \ldots\}$
- $A^{\uparrow} = \{\texttt{RobotMove}(p, q), \texttt{PickObject}(r, o), \texttt{DropObject}(r, o), \ldots\}$
- O: observations
- S: ground states
- A: ground actions
- $\delta: S \times A \rightarrow S$ transition function
- $\rho: S \times A \rightarrow \mathbb{R}$ reward function
- s_0: initial states
- g: (optional) goal states

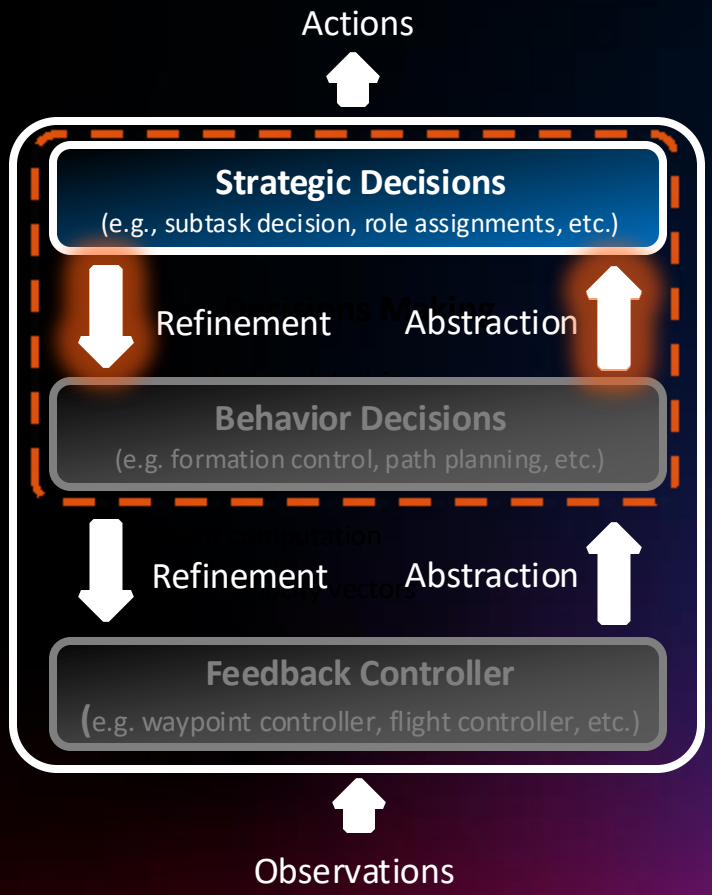


- R. Karia, S. Srivastava, "Relational Abstractions for Generalized Reinforcement Learning on Symbolic Problems, IJCAI, 2022.
- R. Karia, P. Verma, A. Speranzon, S. Srivastava, "Epistemic Exploration for Generalizable Planning and Learning in Non-Stationary Settings", ICAPS 2024.

NS Approach to Decision Making

- ***Example***: In a rescue operation, high-level decisions/tactics are represented as a set of logical predicates (symbolic knowledge) that model both a *single agent behavior* (e.g., "search Area of Responsibility (AOR) A") and the *relations between various agents* (e.g., "drop med kit using UAV X when relative distance to group of people in distress is d and the position UAV Y is P_y")
- Given an RMDP, we can use Deep RL to *learn optimal policies*

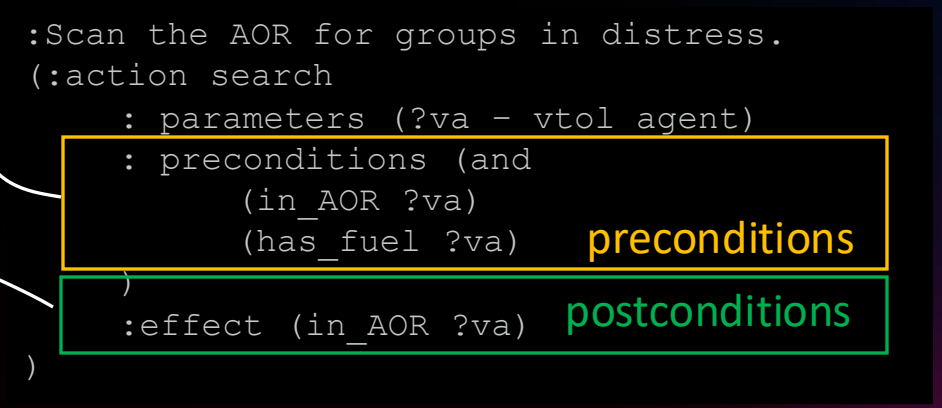




R. Karia, S. Srivastava, "Relational Abstractions for Generalized Reinforcement Learning on Symbolic Problems, IJCAI, 2022.

Behavioral “Contracts”

- *Behavioral contracts*, defined in PDDL, provide a systems engineering approach to formally define a model’s algorithmic capabilities
- Behavioral contracts clearly define the *preconditions* and *postconditions* of AI algorithms making usage terms explicit
  - E.g., preconditions:
    - “VTOL agent in AOR” AND
    - “VTOL agent has sufficient fuel”
  - E.g., postconditions:
    - “VTOL agent in AOR”



- Benefits of behavioral contracts:
  - Assurability
  - Scalability
  - Interpretability/Trustworthiness
  - Compositionality/Hierarchical Decomposition

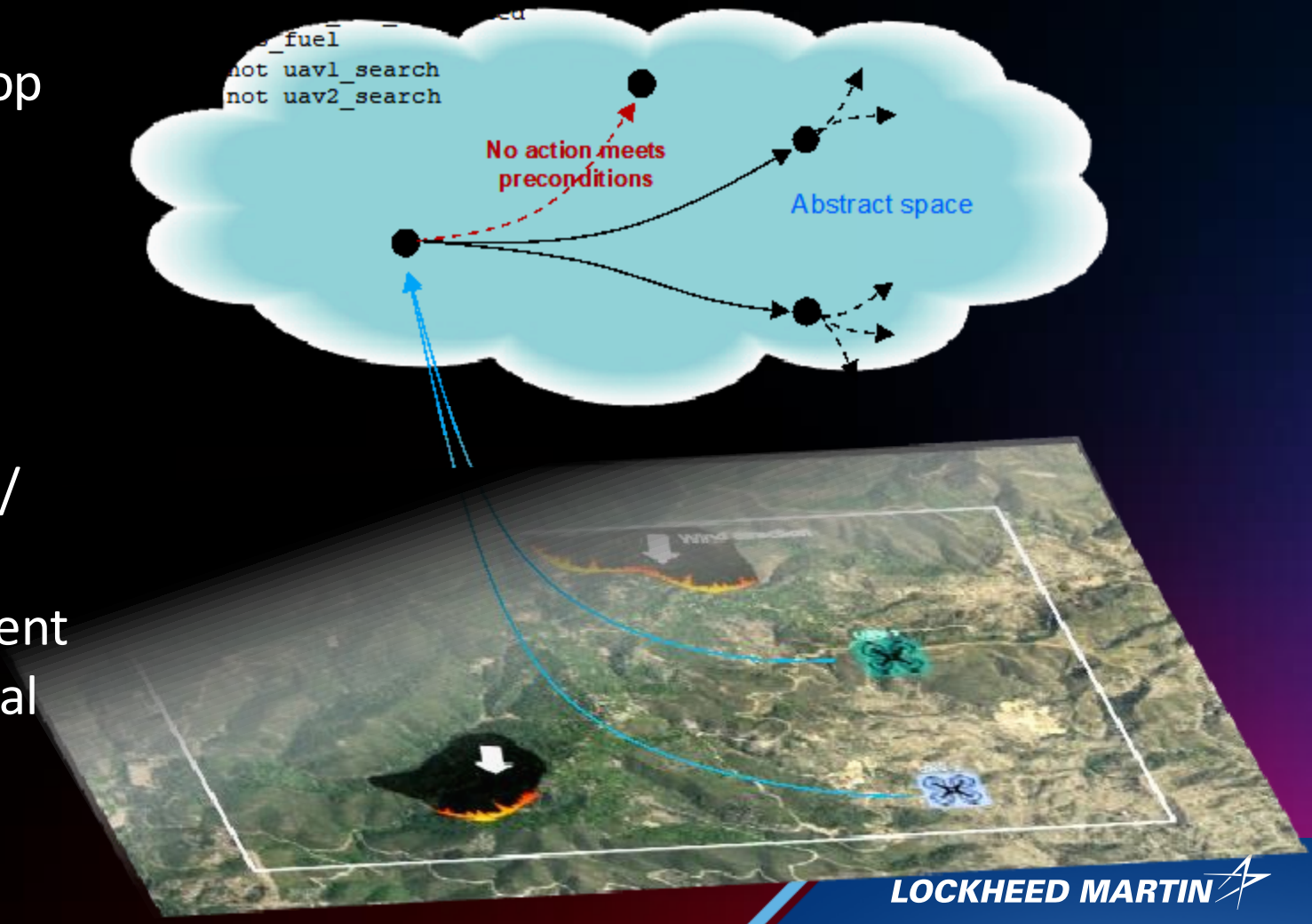
Area of Responsibility (AOR)
Vertical Take-Off and Landing (VTOL)

Relational Abstractions with Action Preconditions

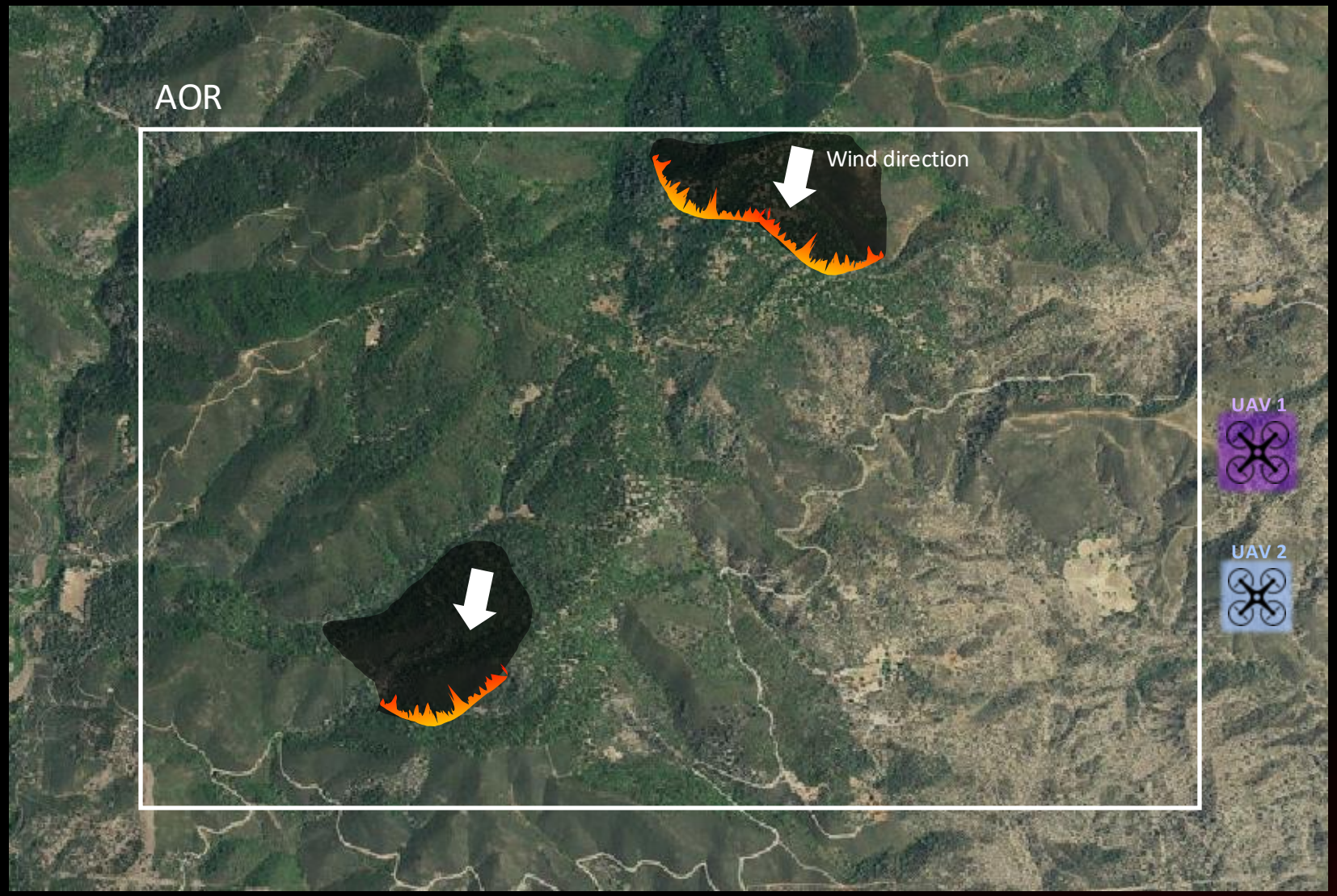
- Provides a way to abstract the problem and generate much *smaller relational abstractions* (neuro-symbolic abstractions)

- Relational abstractions being develop via description logic *provide interpretability* given their symbolic nature

- Modeling requires to define pre-conditions and post-conditions / effects, providing a *compositional framework*, enabling the development of formal assurance cases and ethical compliance artifacts, for MARL



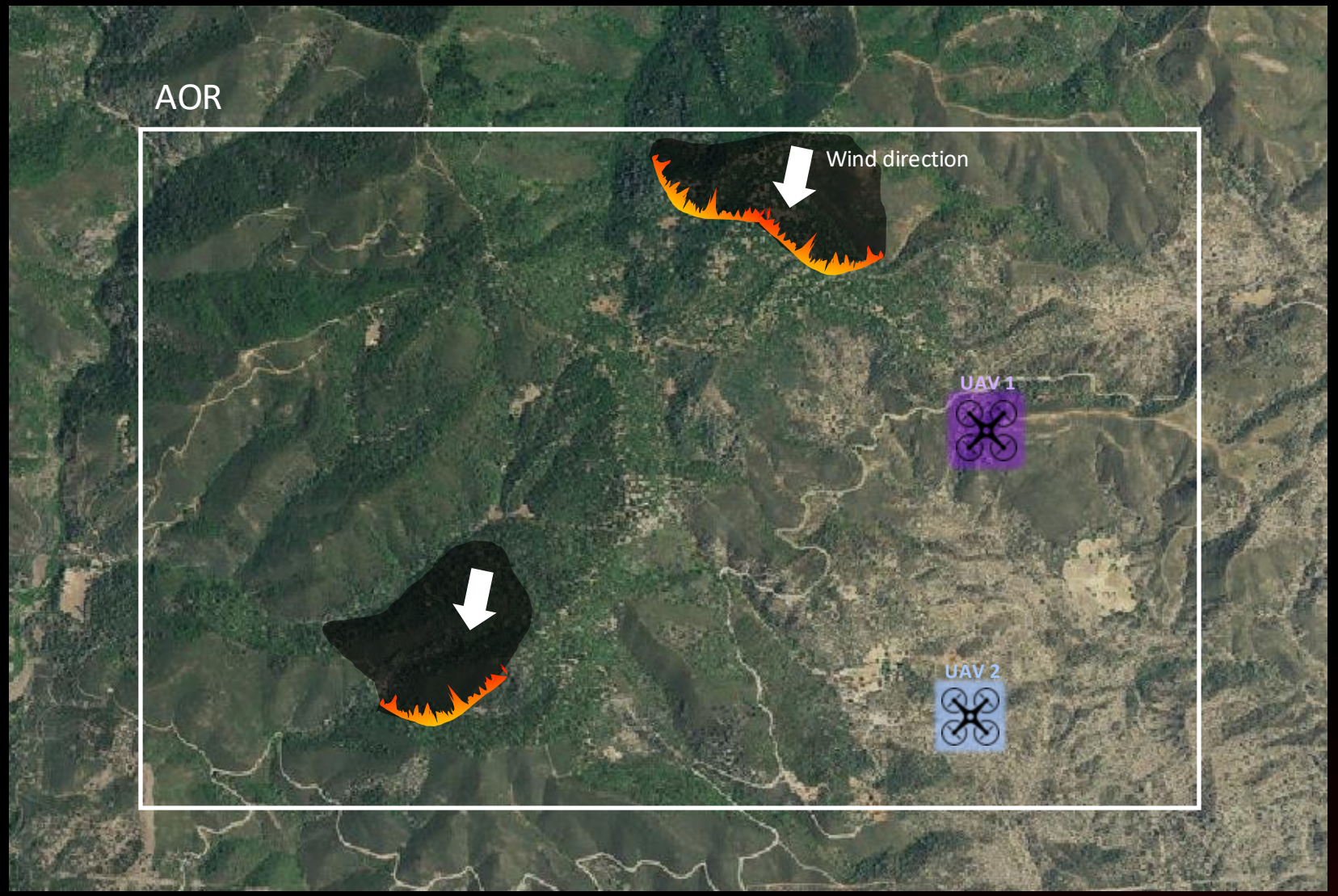
Rescue Operation – Wildland Fires



Agents *Enter* into the AOR (Area Of Responsibility)

State	UAV 1	UAV 2
Inside AOR	False	False
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	False	False
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	False	False
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	True	True
Search	False	False
Error Ellipse Reduction	False	False
Drop Med Kit	False	False

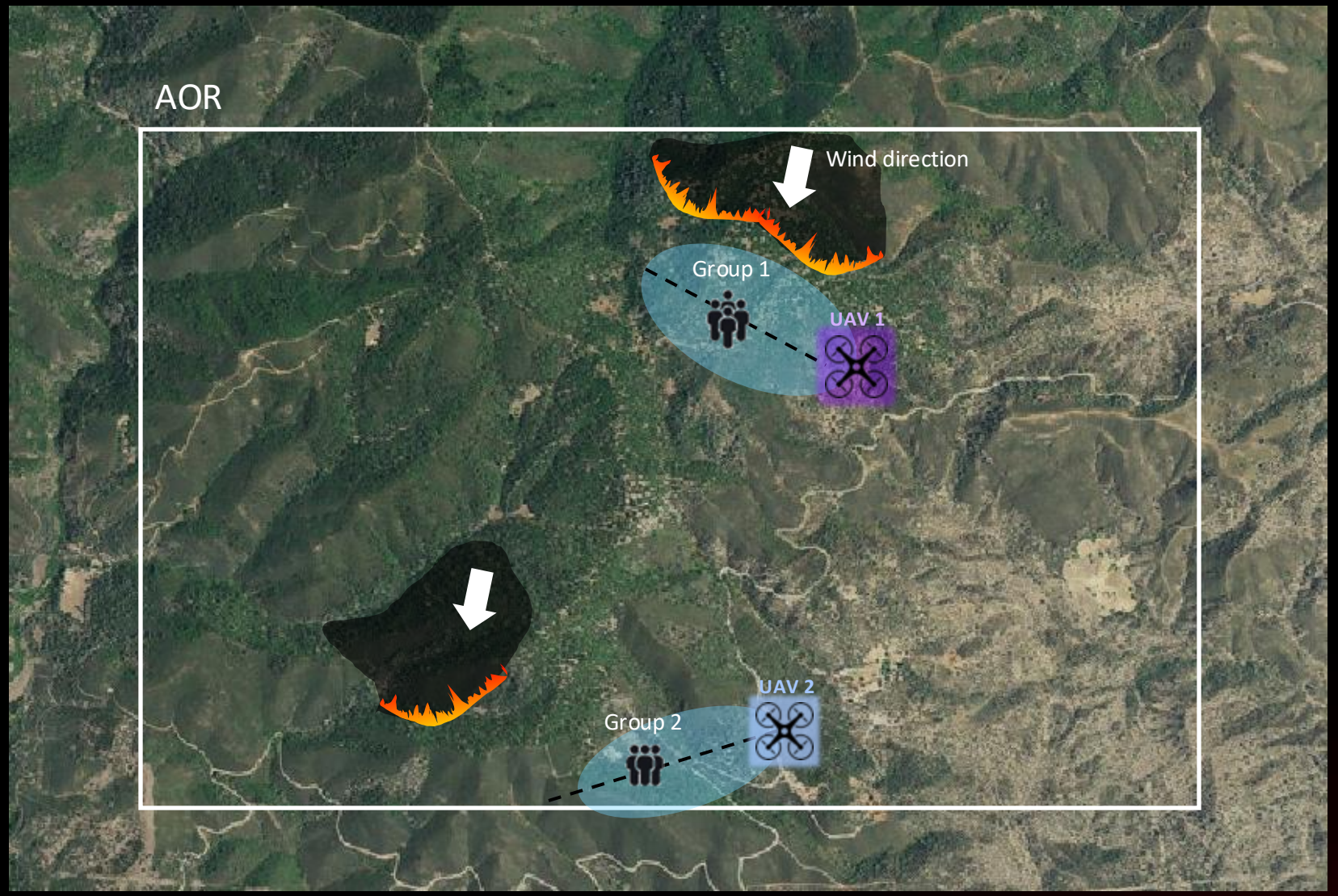
Rescue Operation – Wildland Fires



Inside the AOR the agents start a *Wide Area Search* behavior

State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	False	False
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	False	False
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	True	True
Error Ellipse Reduction	False	False
Drop Med Kit	False	False

Rescue Operation – Wildland Fires

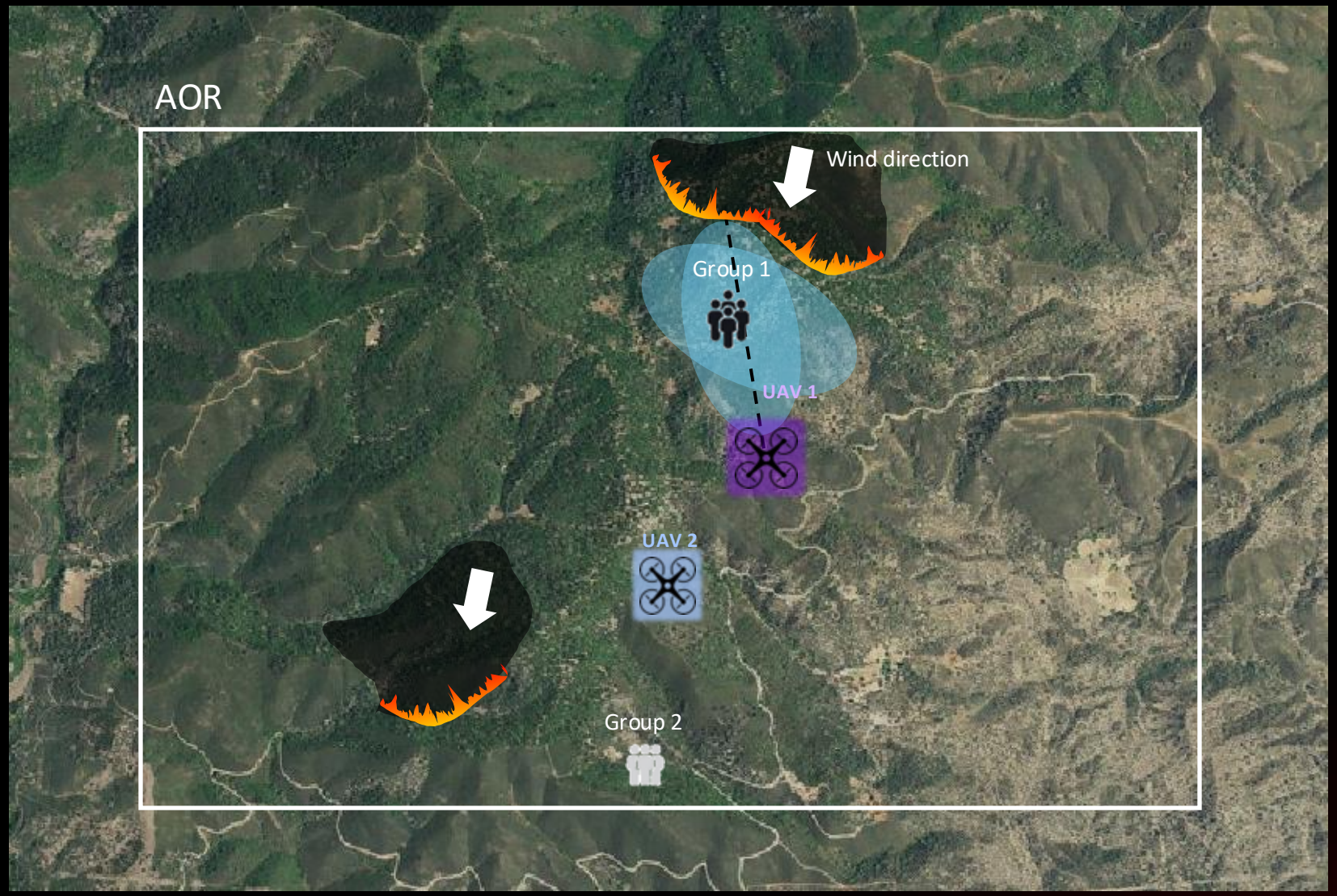


Each agent *detects* a Distressed Group (angle only) and associate an error ellipse

State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	True	False
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	False	False
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	True	True
Error Ellipse Reduction	False	False
Drop Med Kit	False	False

LOCKHEED MARTIN

Rescue Operation – Wildland Fires

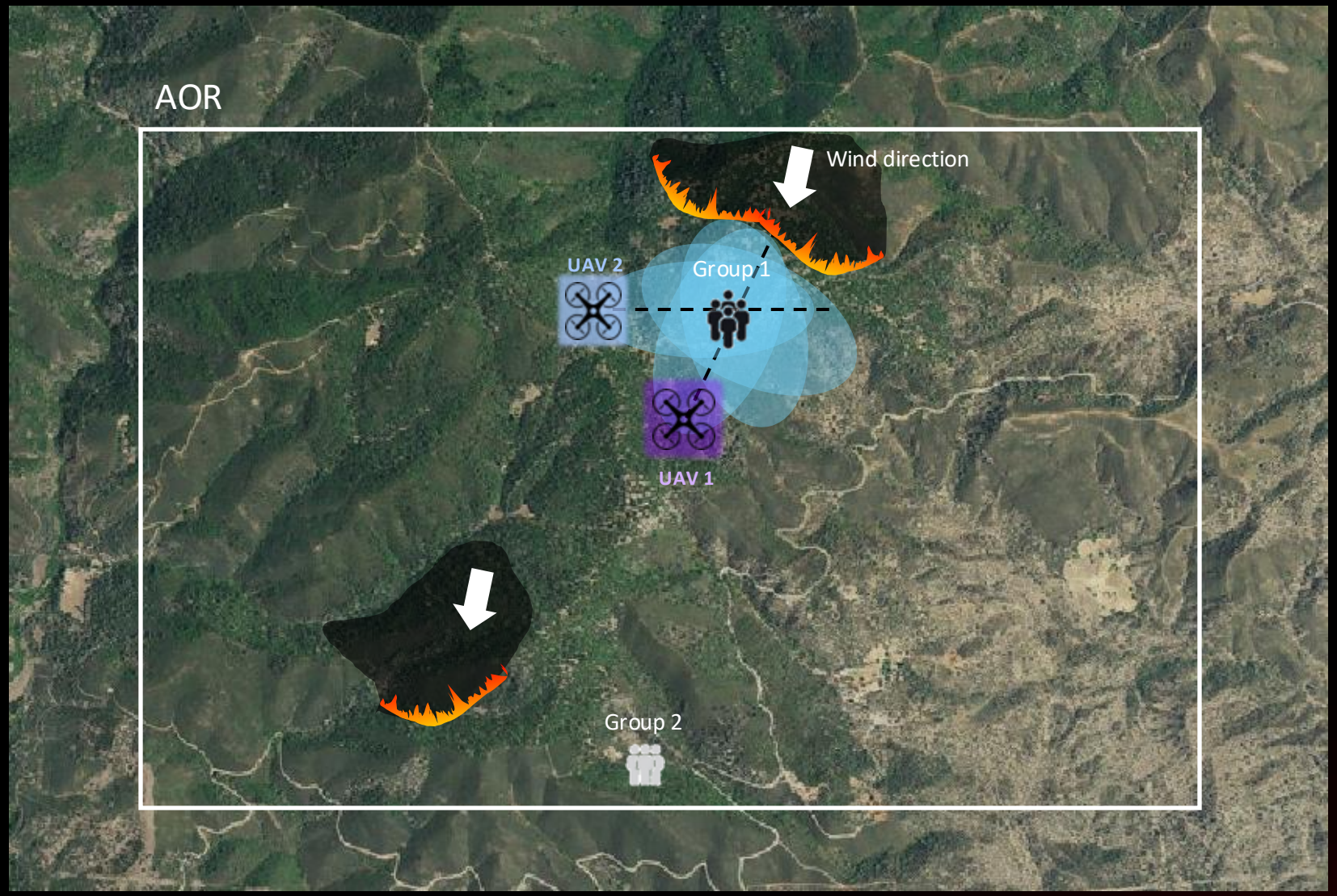


State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	True	True
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	False	False
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	False	True
Error Ellipse Reduction	True	False
Drop Med Kit	False	False

UAV1 continues to reduce error (*Error Ellipse Reduction*) while UAV2 continues searching (Distressed Group 1 prioritized given closeness to the fire)

LOCKHEED MARTIN

Rescue Operation – Wildland Fires

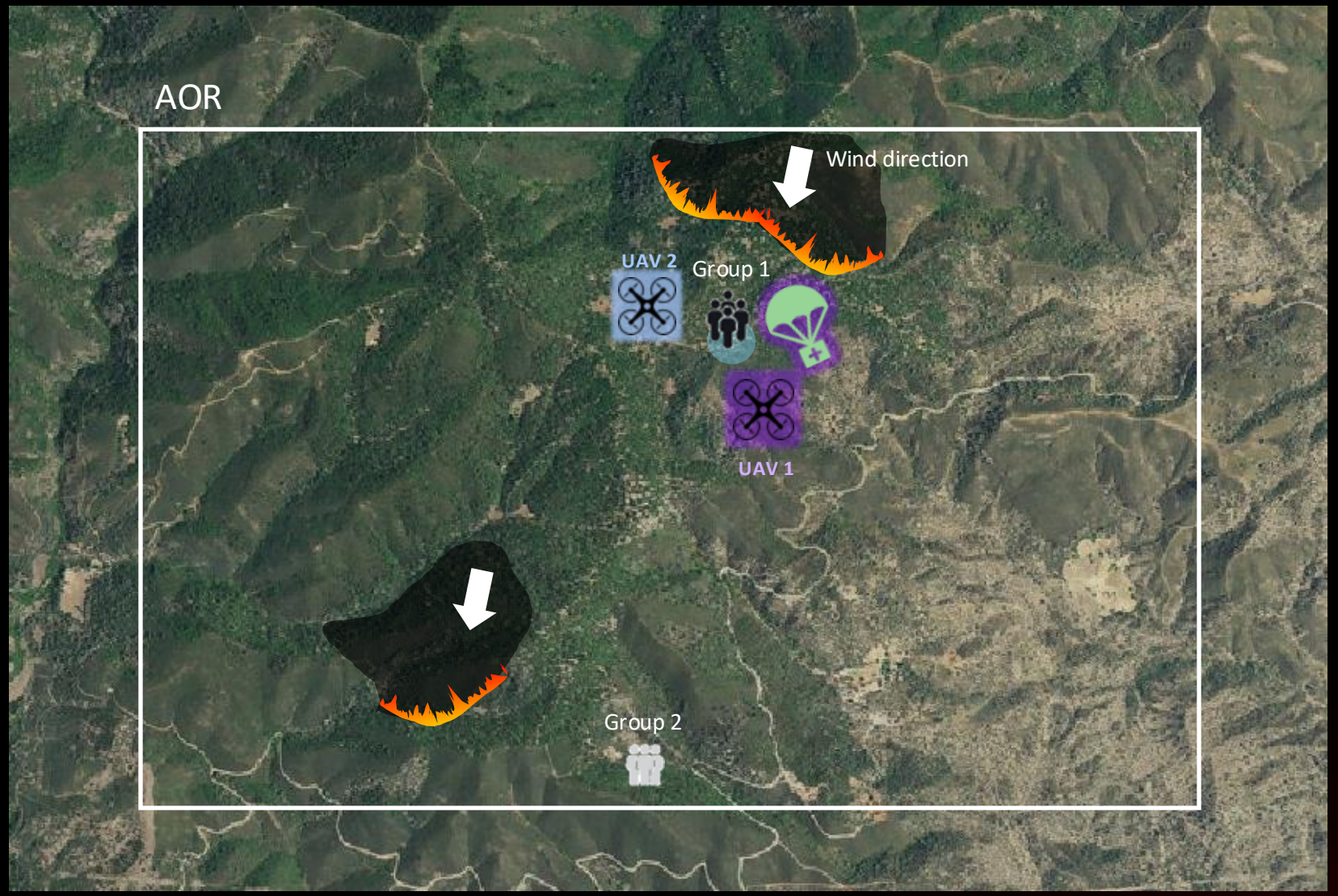


State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	True	True
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	True	True
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	False	False
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	False	False
Error Ellipse Reduction	True	True
Drop Med Kit	False	False

UAV2 is requested to support UAV1 in the *Error Ellipse Reduction* for Distressed Group 1

LOCKHEED MARTIN

Rescue Operation – Wildland Fires



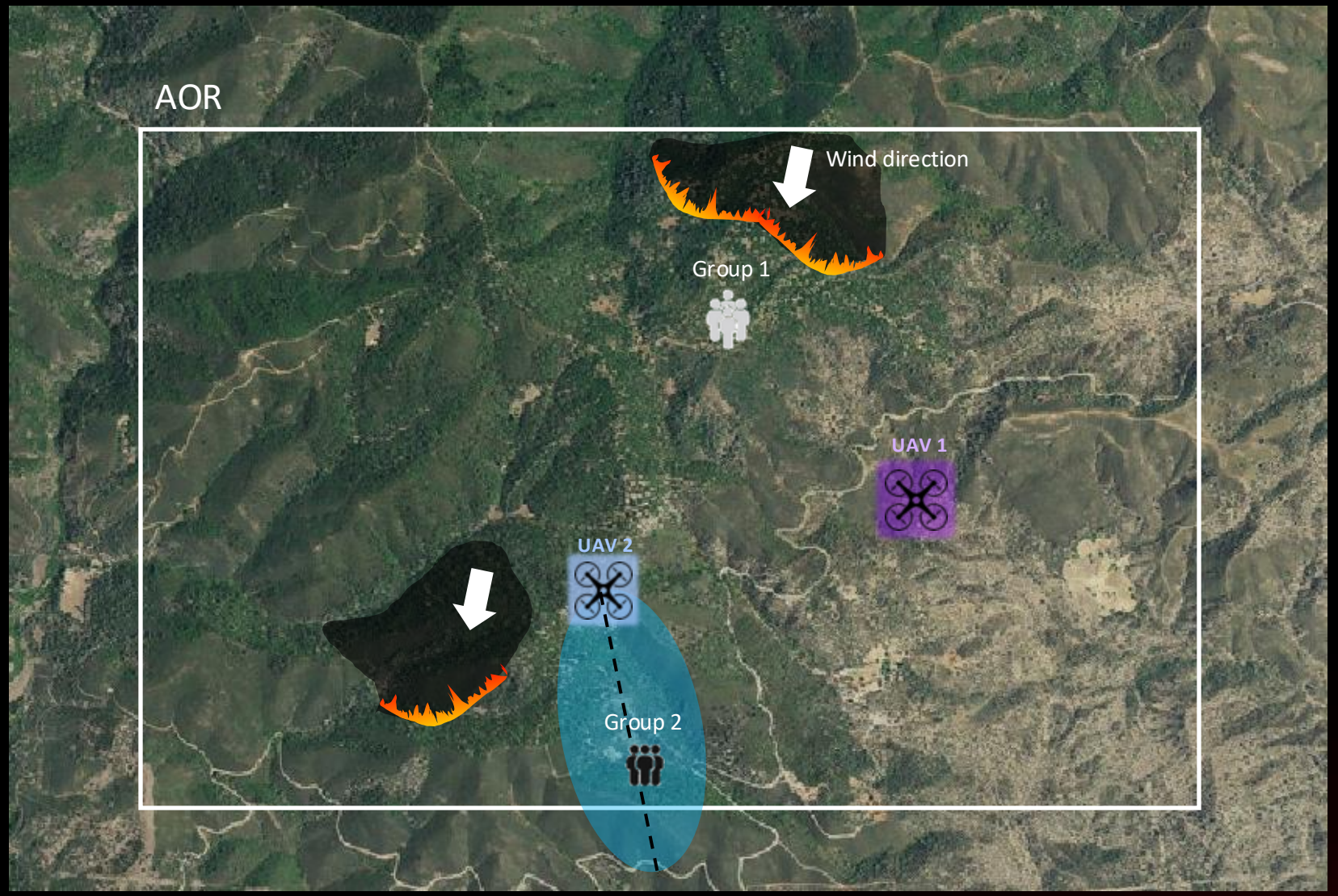
State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	True	True
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	True	True
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	False	False
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	False	False
Error Ellipse Reduction	False	True
Drop Med Kit	True	False

Distressed Group 1 is localized

UAV1, supported by UAV2, *Drops Med Kit* for Distressed Group 1

LOCKHEED MARTIN

Rescue Operation – Wildland Fires

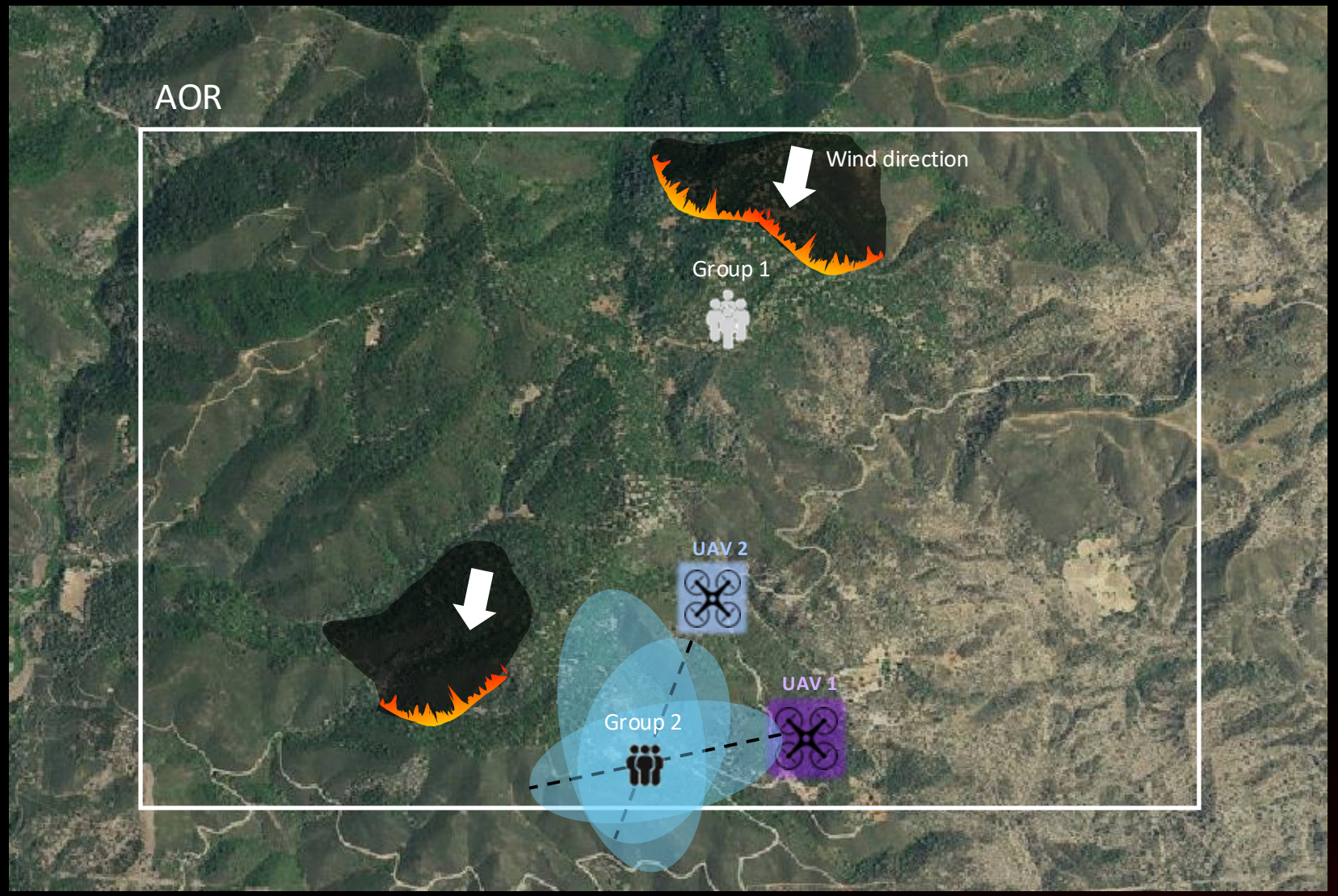


State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	False	False
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	True	True
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	True	False
Error Ellipse Reduction	False	True
Drop Med Kit	False	False

UAV2 is reducing error (*Error Ellipse Reduction*) on position of Distressed Group 2 while UAV1 is searching (*Wide Area Search*)

LOCKHEED MARTIN

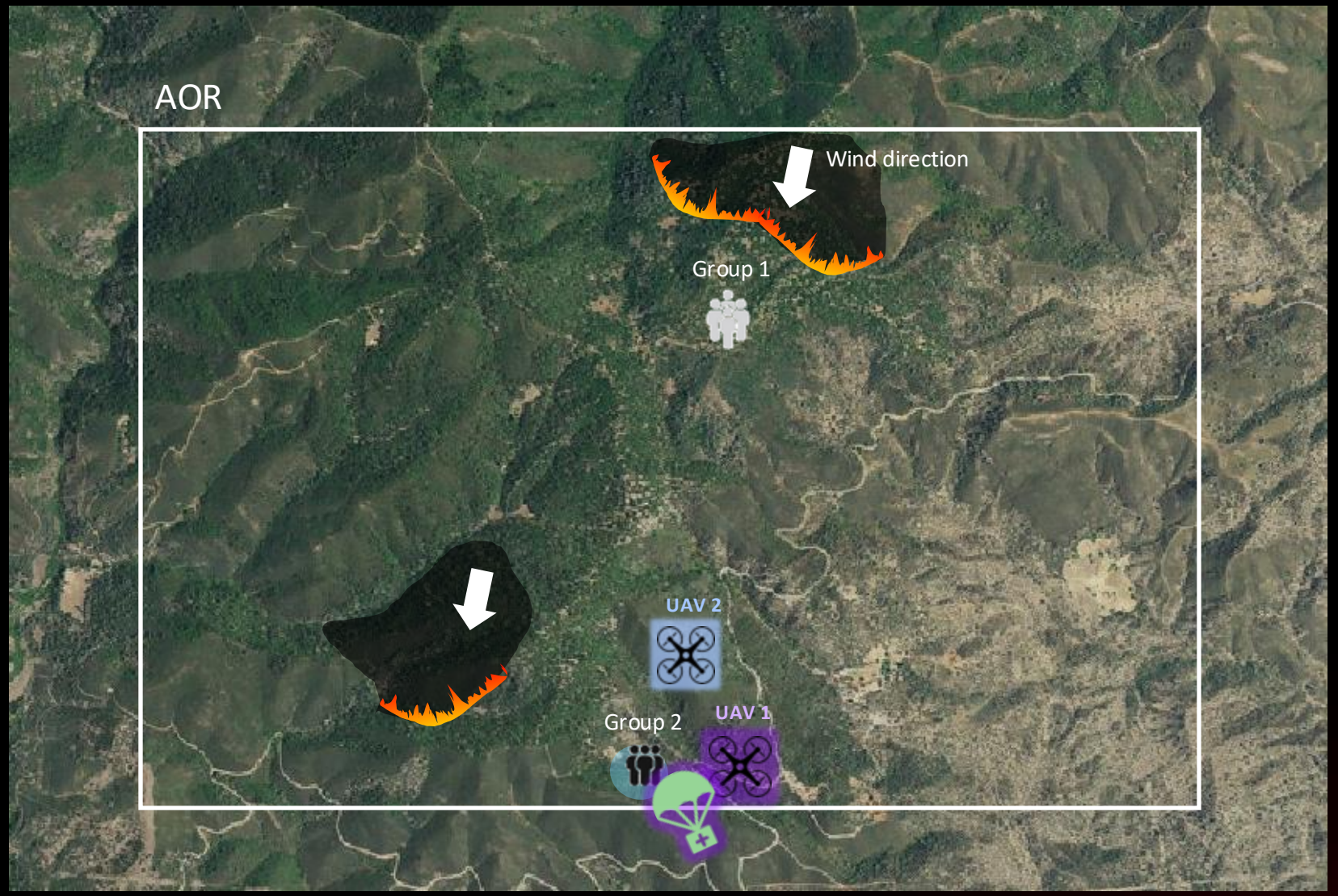
Rescue Operation – Wildland Fires



State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	True	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	True	True
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	True	True
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	False	False
Error Ellipse Reduction	True	True
Drop Med Kit	False	False

UAV1 is requested to support UAV2 in the *Error Ellipse Reduction* for Distressed Group 1

Rescue Operation – Wildland Fires

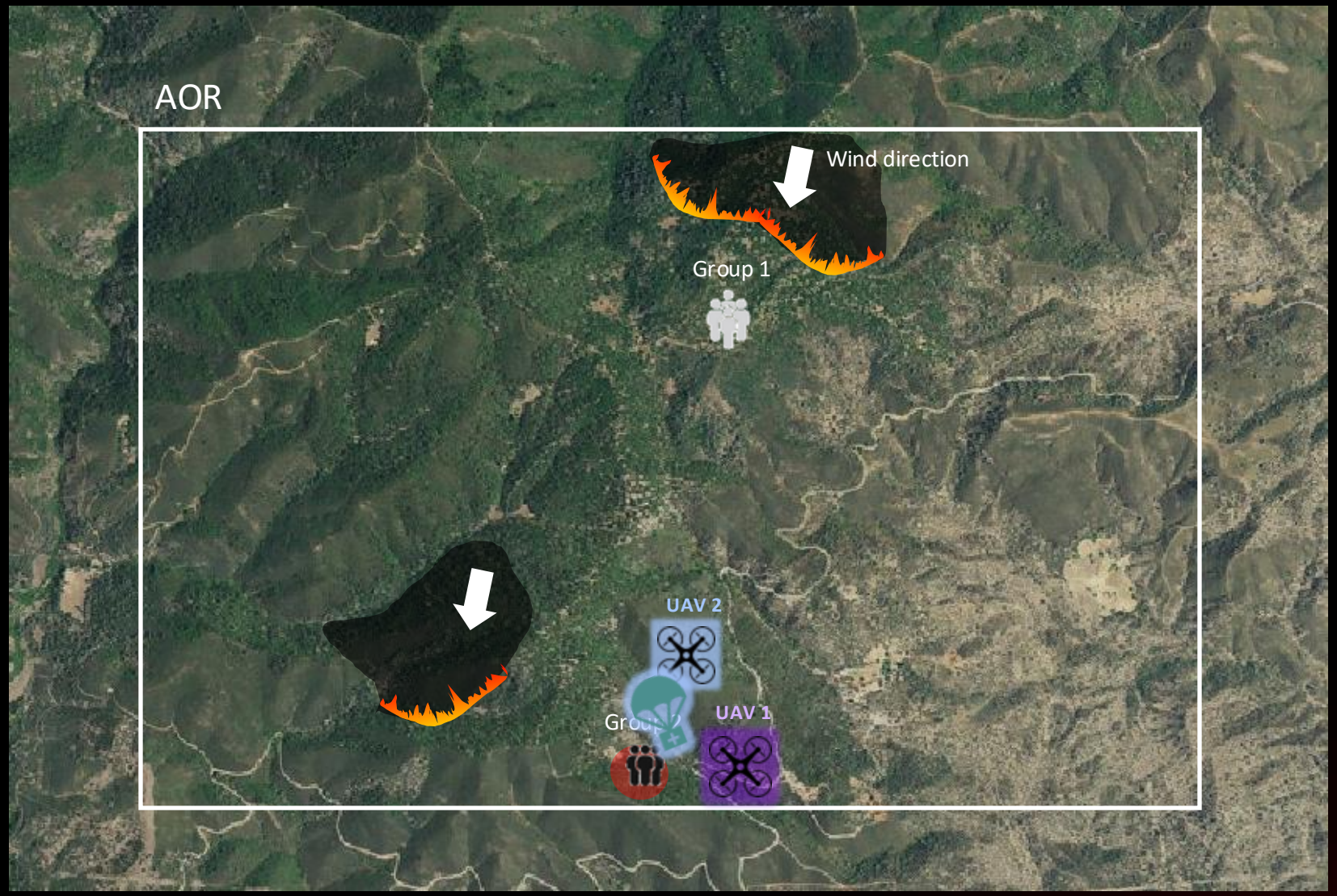


Distressed Group 2 is localized enabling one of the UAVs to *Drop Med Kit*

State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	True	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	True	True
At least 1 Med Kit	True	True
At least 3 Med Kit	False	False
Delivered to Group 1	True	True
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	False	False
Error Ellipse Reduction	False	True
Drop Med Kit	True	False

LOCKHEED MARTIN

Rescue Operation – Wildland Fires

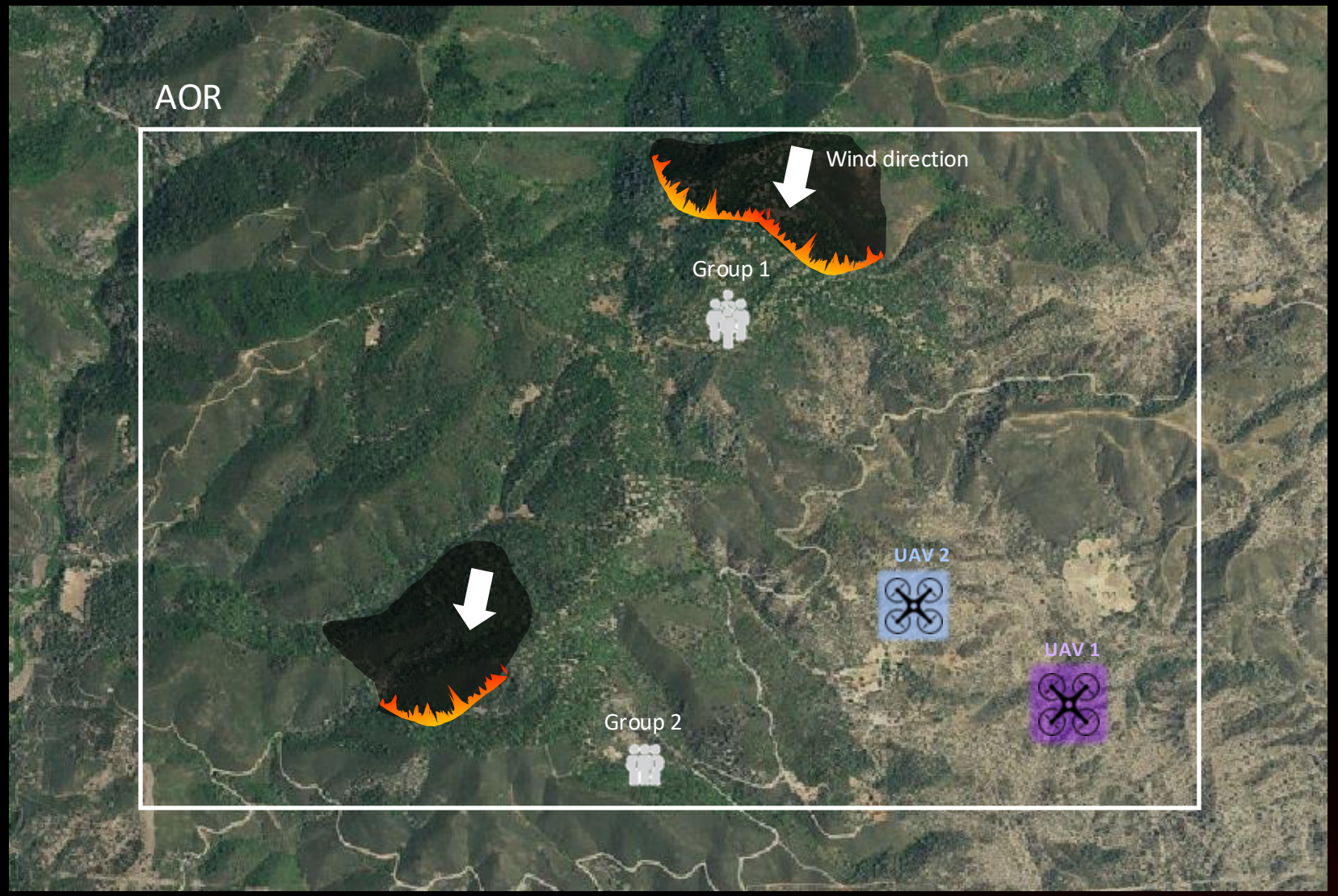


State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	True	True
At least 1 Med Kit	False	True
At least 3 Med Kit	False	False
Delivered to Group 1	True	True
Delivered to Group 2	False	False
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	False	False
Error Ellipse Reduction	True	False
Drop Med Kit	False	True

Alternative Situation: In this case UAV2 *Drops the Med Kit* since UAV1 is out of med kits. UAV1 only supports UAV2

LOCKHEED MARTIN

Rescue Operation – Wildland Fires

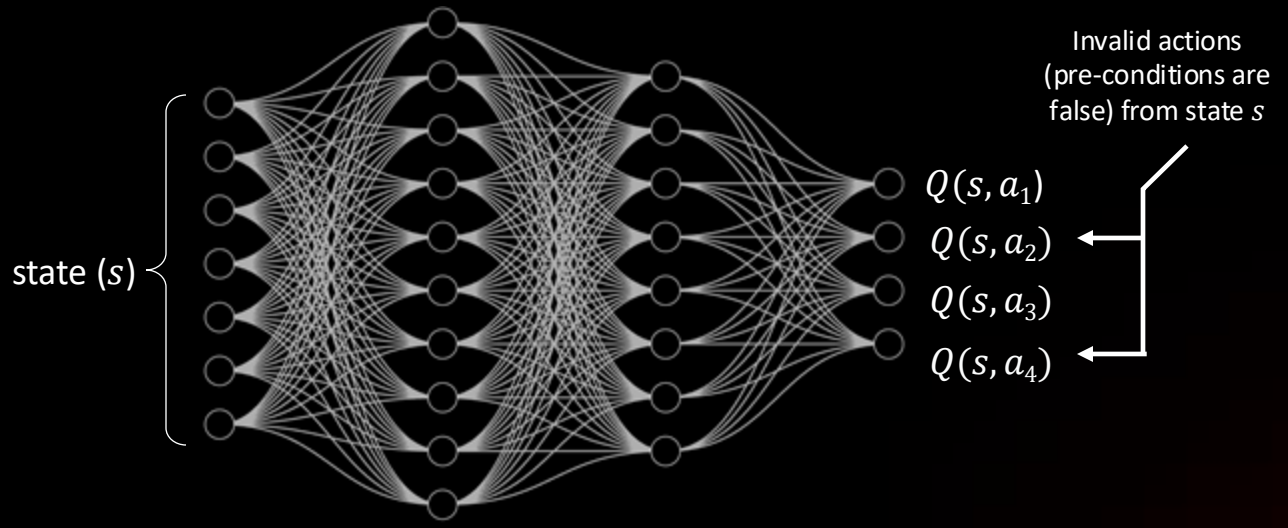


Both UAVs go back to *Wide Area Search*

State	UAV 1	UAV 2
Inside AOR	True	True
Fuel for Search	True	True
Fuel to get to People	True	True
Distr. Group 1 found	False	False
Distr. Group 2 found	False	True
Localized Distr. Grp. 1	False	False
Localized Distr. Grp. 2	True	True
At least 1 Med Kit	False	True
At least 3 Med Kit	False	False
Delivered to Group 1	True	True
Delivered to Group 2	True	True
Action	**UAV 1**	**UAV 2**
Enter AOR	False	False
Search	True	True
Error Ellipse Reduction	False	False
Drop Med Kit	False	False

Preemptive Action Shielding

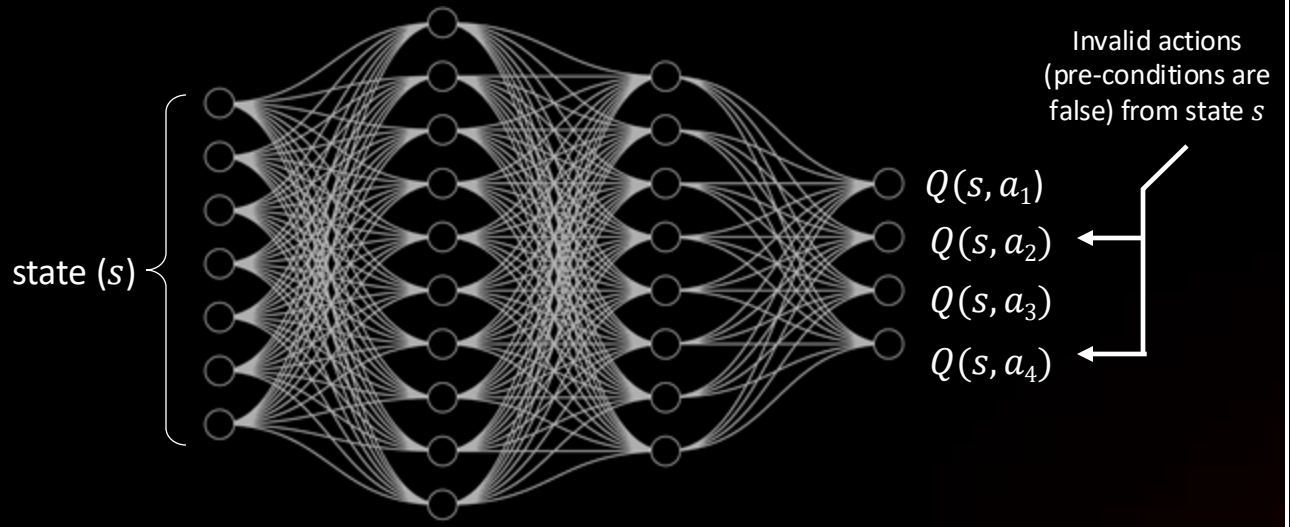
- The process of *action masking* includes everything in the input, and manipulates the output
- DQN is set up by default to take the state as input, and produce a Q value for every action
- Only the Q values of valid actions are considered for action selection



Action Masking: DQN processes all actions at once, only Q values of valid actions are considered

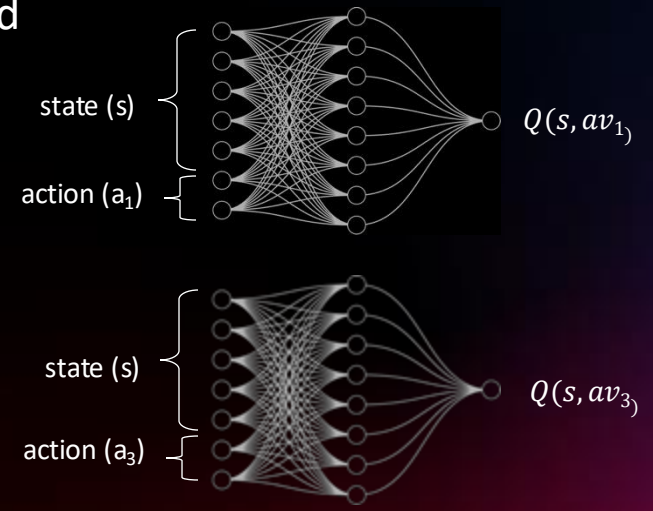
Preemptive Action Shielding

- The process of *action masking* includes everything in the input, and manipulates the output
- DQN is set up by default to take the state as input, and produce a Q value for every action
- Only the Q values of valid actions are considered for action selection



Action Masking: DQN processes all actions at once, only Q values of valid actions are considered

- Unlike action masking, *preemptive shielding* enforces that only actions that satisfy its preconditions are considered by the network
  - This is not only useful for safety and assurance purposes but also to prevent algorithmic deadlocks in decision making
- DQN is modified to take as input the state and a valid action and return a single Q value. This process is repeated for every valid action
- This process is slower than traditional DQN, but the small size of the predicate action space mitigates this loss in speed



Preemptive Action Shielding only feeds valid actions into the deep Q network

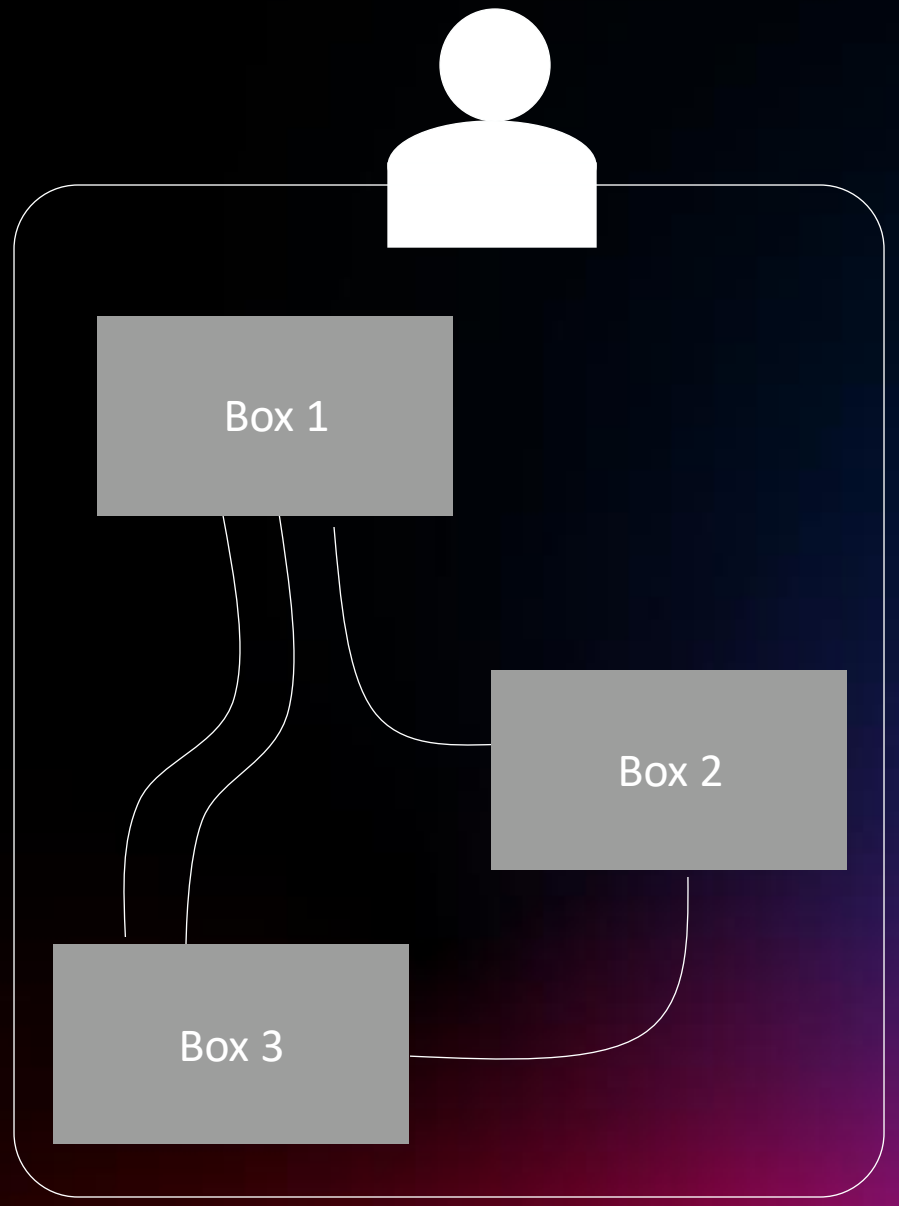
Some Important Challenges

What is the “right” architecture?

Today:

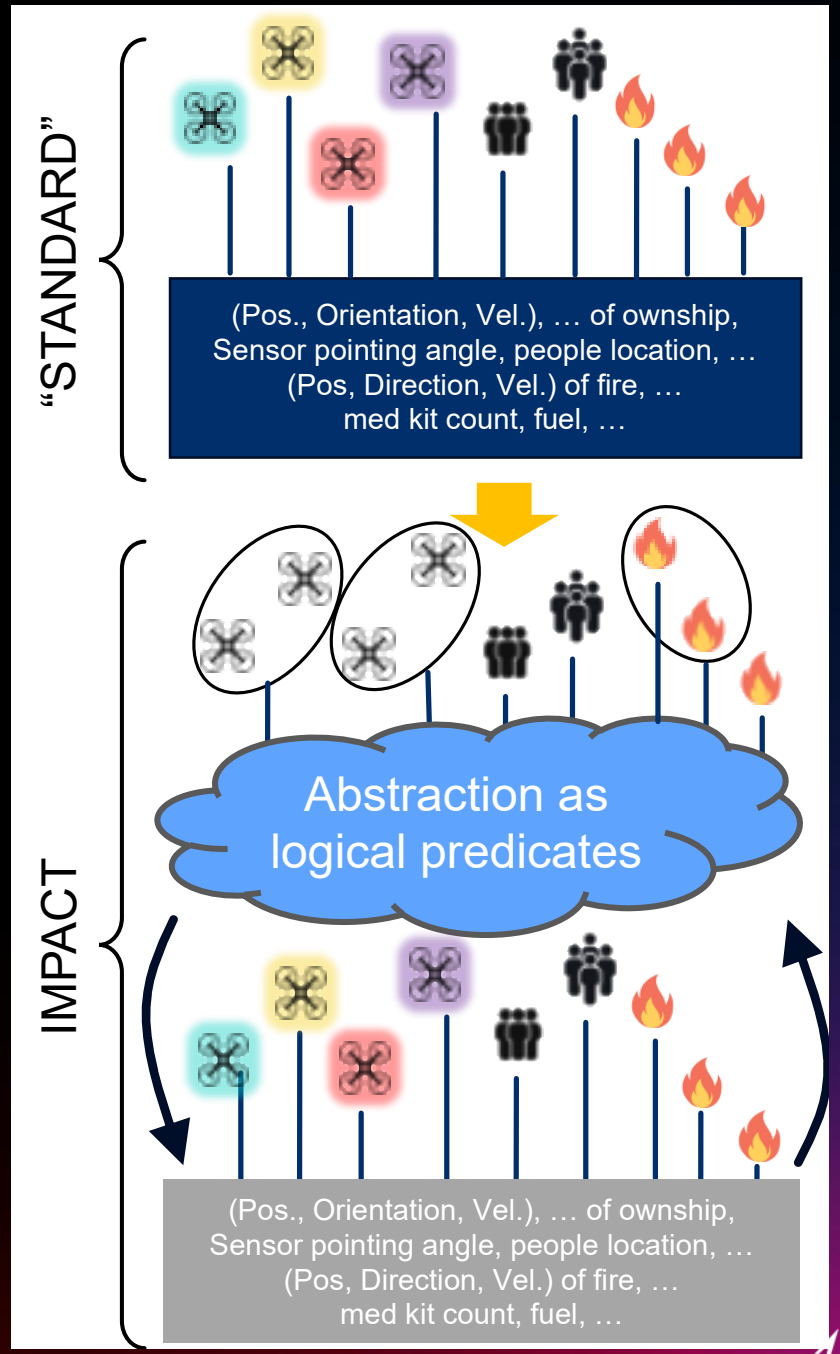
- They re based on a set of *requirements* that took a lot of iterations between different groups to emerge
- They are based on a *“historical” decomposition* of functionalities that took a long time to arise
- They strongly rely on the *presence of humans* to deal with all the “forgotten” requirements and issues

- The *lack of an architecture theory* constraints the design to follow “old” paradigms that may not be suitable when considering today/tomorrow's complex systems (e.g., autonomous systems with ML-based components)



Summary

- NS approach for multi-agent decision making that is:
  - Scalable as the number of states and actions increase
  - Flexible so that we can consider heterogenous capabilities
  - Assurable so that we have guarantees about its behavior
- Relational abstractions enable to “lift” the problem to a smaller space
- Explicitly express pre-conditions and effects of each behavior/tactic



Conclusions

- The OODA loop provides unique challenges for NS architectures but they also provide key opportunities to design trustworthy autonomous systems
- Knowledge capture and its symbolic encoding into NS architecture will require further research to capitalize on the benefits of the symbolic and subsymbolic “worlds”
- Composition, interpretability and reusability of NS abstractions in the OODA loop can unlock new and more scalable design paradigms
- Uncertainty and ambiguity stemming from the “compression” to a symbolic world will need to be carefully managed
- Assurance of autonomous systems can strongly capitalize from NS architecture

