Detection of Complex Events in Synthetic Aerial Sensor Data with NeuroSymbolic Reasoning

Henry Phillips[1], Mani Srivastava[2], Ben Purman[1], Jeff Craighead[1], Brian Wang[2], Julian de Gortari Briseno[2], Lance Kaplan[3]

1 Soar Technology, Inc
4715 Data Ct, Ste 400
Orlando FL 32817
(734) 627-8000
henry.phillips@soartech.com
ben.purman@soartech.com
craighead@soartech.com

2 University of California at Los Angeles (UCLA)
56-125B Engineering IV Building
420 Westwood Plaza (Box 951594)
Los Angeles, CA 90095-1594
(310) 267-2098
mbs@ucla.edu
wangbri1@g.ucla.edu
julian700@g.ucla.edu

3 DEVCOM Army Research Laboratory (ARL)
2800 Powder Mill Rd,
Adelphi, MD 20783
(301) 394-0807
lance.m.kaplan.civ@army.mil

Abstract

To achieve overmatch in C4I versus peer adversaries, intelligence analysts reviewing sensor feeds need help in identifying and classifying instances of adversary tactics, techniques, and procedures (TTP) execution quickly and accurately across environments and domains. Detecting a few relevant entity movements or clues from among a huge haystack of other contacts is extremely difficult, particularly when those clues may be observed by different sensors, or separated in time by minutes or longer. TTPs also change frequently, and relatively little training data may exist to help a system learn to detect and recognize TTP instances. An approach is needed that can associate TTP indicators from across the wide range of sensor data, while also addressing these limitations on training data.

This work describes NeuroPlex++, an expansion of a previously developed tool, *NeuroPlex*, which frames TTP detection as a Complex Event Processor. A key innovation is the use of a hybrid, neurosymbolic architecture to integrate data with subject matter expert inputs. This architecture uses deep neural networks to efficiently map unstructured high-dimensional sensor data into symbolic percepts embedded in space and time. A logical layer allows for human-encoded or system observed knowledge incorporation in a machine learning pipeline. To demonstrate our approach, we rely on a flexible, synthetic data testbed, and we instantiate hundreds of randomized instances of these TTPs intermingled with synthetic clutter vehicles. We show detection results of NeuroPlex++ operating on sets of video feeds generated with this synthetic data tool on realistic adversary TTPs.

Keywords: neurosymbolic reasoning, complex event processing, data synthesis

PROBLEM SUMMARY

In a future fight against a peer adversary, speed of decision-making and well-grounded situational understanding are critical to battlefield effectiveness. The massive increase in airborne sensor capability provides an opportunity to use sensor data to make quick and effective decisions, provided techniques can be developed for helping ISR analysts manage the overwhelming volume of data produced. ISR analysts need tools to integrate disaggregated data into a coherent picture of enemy activity, piecing together disparate, atomic events and activities into more complex events that indicate higher-level activities, like tactics, techniques, and procedures (TTPs). *Maintaining tools to recognize TTPs in a dynamic battlefield environment is particularly challenging because they change on short notice*. While data-driven analytics are important, a fully data-driven approach will only be successful at recognizing yesterday's TTPs because of its dependency on available training data. What is needed is an approach that combines state of the art data-driven ML techniques with adaptation using human expertise to *recognize* complex events and TTPs in a dynamic, battlefield environment, and to *quickly update the library of these TTPs*.

Recognizing known TTPs is the first challenge. Complex Event Processing (CEP) [Luckham02] provides a mechanism for the detection of real-world, complex events. The research community has made significant progress recently in combining deep learning architectures with human domain expertise to recognize complex events across sensor modalities using neurally reconstructed logic and arithmetically differentiable circuits [Ahmed22] with minimal re-training, which can be used to enable *adaptation to domain shifts* indicative of new TTPs. These TTPs can be modeled as a set of complex events that are comprised of many atomic events distributed over a wide range of time and space, with those atomic events defined by the positions and movements of entities and objects. Table 1 provides a list of definitions levels in such a taxonomy spanning objects to events to TTPs.

Event Type	***Definition***
Object Detection & Tracking	• Event involving the detection of position and movement of a single entity
Atomic Event (AE)	• Event involving watchbox/tripwire and position detection or separation among multiple entities • AEs are intended to minimize state information (over short intervals) • AEs are defined by the location and movement of objects/entities
Complex Event (CE)	• Most narrowly defined event that would have tactical relevance to an intel analyst witnessing it • Stateful, capturing change over time • Requires stitching info together from multiple sensors, and over a longer window of time • CEs are made up of AEs
Tactics, Techniques, & Procedures (TTP)	• Force level methods for accomplishing military goals • TTPs are made up of CEs

Table 1. Event Level Definitions

We next consider how typical sensor processing approaches and multi-domain sensor fusion inform this CEP challenge, as well as strategies for generating useful data, a critical requirement for such a complex event processor.

Data-driven machine learning, particularly deep learning, is the foundation of the current state of the art in processing raw sensor data. Open source solutions exist for image classification, object detection, instance segmentation [Wang22] and activity detection [Heilbron15], and open-source tools are readily available to integrate these object-level detections into tracks [LearnOpenCV, n.d.]. These methods are highly related to current sensor processing methods and data fusion architectures, which are optimized for processing large volumes of multi-sensor data. These methods are useful in building complex event processors, *but they are focused on timescales that are insufficient for recognizing complex events spanning more than minutes or across sensor domains* [Herath17]. The approach described here incorporates the strengths of both approaches, relying on DL for atomic event detection, and on CEP to recognize complex events spanning greater time and distance.

The **Complex Event Processing** [Luckham02] literature is similar to that of data fusion, in that complex events are comprised of atomic events, though both types can be comprised of spatial, temporal and other features. As defined here, complex events are those sufficiently aggregated to have tactical relevance to an analyst, and that can be further subsumed by enemy tactics, techniques, and procedures (TTPs). A TTP is comprised of a range of coordinated actions, but crucially for modeling, how these actions present themselves in the environment is highly variable and dependent on specific terrain, weather, sensors, sensing geometry, personality/behaviors of those participating, and the interactions of all of these components.

Variations in TTP instances increase the *amount of data required for data-driven approaches* for CEPs to achieve desired accuracy, and they underpin the lack of performance in data-driven approaches for long duration activity detection. *Volume of data* with relevant complex events for multi-domain operations (MDO) in the military domain has been a key limitation for developing and evaluating this technology for use in DoD applications. The data itself is expensive to collect due to the number of assets (vehicles, people, sensors) required, and data labeling is a well-known limitation in applying ML algorithms.

The lack of available suitable data remains a key limitation in developing CEP methods for this domain. Existing datasets do not exist that capture multi-camera views of adversary tactics, separated over relevant time and space and encompassing large numbers of people and vehicles. **Synthetic data** is an appealing solution to the data availability problem, but a dataset must be sufficiently relevant and realistic to be useful. Deep learning methods are exceptionally good at learning to process synthetic data, but this performance doesn't necessarily translate into real-world performance. Synthetic data generation tools have been demonstrated to improve real-world performance of object detectors [Craig21] and reliably predict real-world performance as a validation dataset [Martinson21]. Sensor and environment data synthesis must incorporate **perceptual realism**, the degree to which the level of detail captured in sensor feeds are modeled realistically, as well as **behavioral realism**, which means that the entities within the simulation are executing behaviors within the constraints of the TTP, but also exhibiting realistic variability. That is, enemy vehicles and people need to maintain a realism for how they maneuver within the terrain, relative to each other, and within the scope of specified TTP. Neutral entities, or pattern of life entities, also need to exist in the environment, and the way that enemy vehicles interact with neutral entities needs to be realistic. Lastly, the sensors themselves, and how they collect data, need to be consistent with expected data collection behaviors. These datasets must incorporate both positive and negative examples of AE/CE presence in the training data. To provide sufficient quantity and variability, the synthetic data must contain variable iterations of scenarios involving execution of targeted AEs and CEs along with negative examples for both red entities and background/pattern of life (POL) entities, yielding ground truth logs for both event occurrence and entity placement at the video frame level.

These data-intensive approaches must also address the problem of resilience to *changes in those TTPs* [Tenzer22] which would demand immediate incorporation into CEP models. This means that one of the

most important functions of such a system would be the ability to adapt to new tactics and environments quickly, and *without large amounts of additional training data*. This can be achieved through the introduction of neurally reconstructed logic (NRL) into the CEP, to capture SME information in symbolic form, restructured and quantified using arithmetic circuits, and then use it as the basis for retraining neural layers to recognize the new events captured in the SME input. This translation of human expertise must be managed by the use of a customized CEP grammar for direct incorporation of domain expertise in symbolic information coded into arithmetic circuits to be used to retrain neural components *without new data beyond what the SME provides,* in order to ensure the detection system remains useful in the face of rapidly evolving enemy TTPs.

PREVIOUSLY DEVELOPED CEP APPROACHES

Complex event processing (CEP) refers to a set of computational techniques with roots in time-series databases and programming languages but now used in the Internet of Things (IoT) settings and cyber-physical systems (CPS) as well, developed for processing incoming real-time events to extract meaningful information. The incoming events indicate some activity or change in the state of the world and could be in various forms, such as a sensor reading or an SMS message. Incoming events are analyzed and correlated to discover and infer *complex events*, which are rigorously defined in some formal language as patterns of events involving both sets of events and relationships between events, such as timing, location, and causality. Under the hood, state-of-the-art CEP systems use efficient pattern matches that scale to high-speed event flows and can detect patterns of multiple events occurring over long periods of time. This process, referred to as complex event recognition (CER), may also incorporate uncertainty and approximate matching of event patterns [Alevizos17]. For example, a distributed attack on a cyberinfrastructure could be inferred via a pattern involving suspicious events at multiple network nodes within a short time interval. Likewise, a hospital may do CEP over sensor data to detect and remedy health safety violations. Typically, the detection of a complex event in turn triggers downstream actions by a human or by an automated system. Moreover, the complex events may be organized in an event abstraction hierarchy. Relevant to the proposed research, a key limitation of state-of-the-art CEP systems is that they are designed to work with input events that are structured and low-dimensional. So while they work well for enterprise applications with events such as a temperature going above a threshold, a customer entering a website, a person leaving a building, etc., they cannot directly infer complex events from unstructured and high-dimensional data such as video streams, natural language text, acoustic sensor data, etc. Also, these systems depend on expert-designed event patterns and cannot learn from data. While potential methods for inductive logic programming (ILP) [Law20] could be extended to learn event pattern rules for CER from examples of input event sequences and corresponding complex events, current ILP methods do not handle time and location as first-class entities.

***Neural approaches to detecting complex activities and events*:** In deep learning, recurrent neural networks such as LSTM are used to monitor temporal data for purposes of event detections and activity recognition. However, there are limitations that make just using LSTM (or other recurrent neural networks) to predict complex events not a good approach. First, modeling long-term dependencies requires memory and with typical sensor sampling rates can grow to large amounts and also require huge volumes of data to learn from. As a result, even large RNN models are limited to a few hundred timesteps and a few seconds with modalities such as video and audio [Singh16, Cakir17]. Even architectures such as Temporal Convolution Networks (TCN) [Lea16] and Transformers with attention mechanisms [Dai19,

Zhou21] do not help much with memory reaching ~10s and ~1K steps. Besides the challenge of limited temporal memory, the purely neural approaches also suffer from models that are not inherently interpretable, requiring posthoc explanation methods that generally work by highlighting salient features [Chattopadhay18, Lundberg17], providing approximate local models that are interpretable [Ribeiro16], or offering relevant examples from the training set [Jeyakumar20]. Moreover, the interfaces between layers that are the components of end-to-end deep learning models are generally not human-comprehensible, which makes it difficult to reuse them in the way symbolic algorithmic approaches allow.

***Neurosymbolic approaches to detecting complex activities and events*:** In research conducted under the US-UK DAIS ITA program (https://dais-legacy.org), we introduced neurosymbolic architectures for CEP [Vilamala20, Vilamala21, Vilamala23, Xing19, Xing20] that combine symbolic reasoning and neural representations to create a whole that is greater than the sum of its parts. The intuition was that differentiable and over-parameterized neural components learnt from data can efficiently process sensory inputs to create precepts to assist symbolic reasoning and make it scalable, while the symbolic components expressed as logic rules can provide interpretability and analyzability, enforce constraints at runtime, allow for injection of human knowledge, and act as regularizers that guide the learning of neural components. We investigated several variants with different characteristics as shown in Figure 1. Starting from a stream of raw sensory data, potentially from distributed multimodal sensors, neural networks are used for summarizing the sensory perception in a form that can be digested by either a differentiable probabilistic logic program allowing for the gradient to be back-propagated [Vilamala20] or by another neural network that has been trained in a teacher-learner fashion [Hinton15] and then frozen while training the entire pipeline [Xing20]. Other research in the DAIS-ITA, while not in the context of CEP, showed that neurosymbolic approach also offers advantages of robustness to domain shifts [Cunnington21].

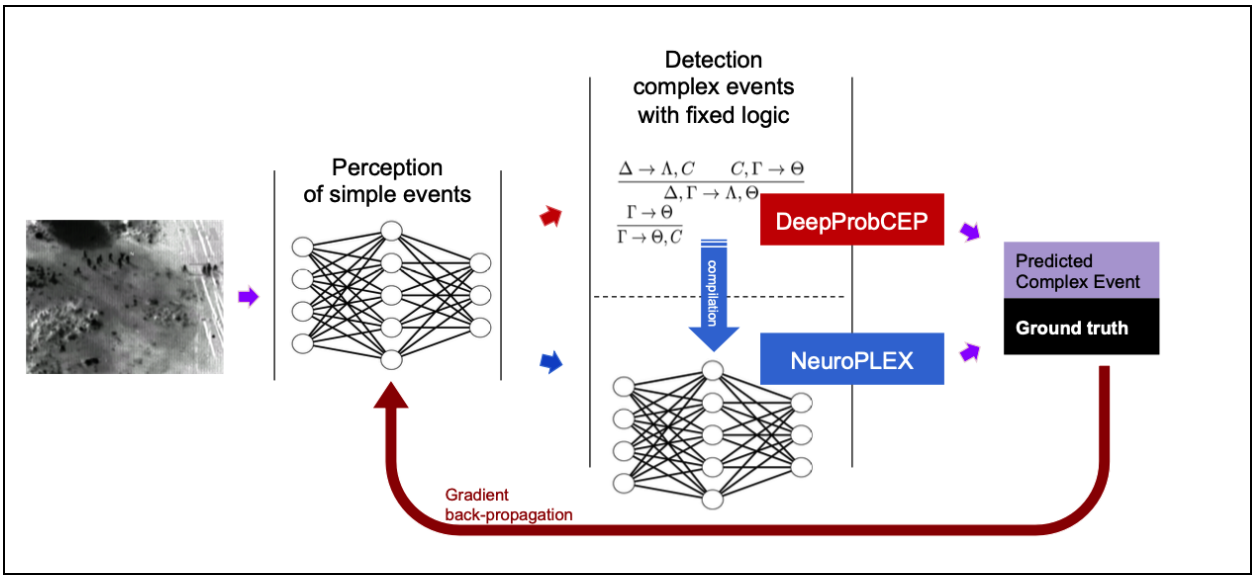

Figure 1. Neuroplex and DeepCEP Neurosymbolic Architectures for Complex Event Processing

While providing promising results [Vilamala20, Vilamala21, Xing19, Xing20], the neurosymbolic paradigm for complex event processing in the initial research is restrictive and inflexible: it uses neural networks to map high-dimensional data to a fixed set of pre-defined low-level concepts or events [Jeyakumar23], which are then centrally analyzed symbolically by an extensible set of probabilistic logic rules that have no concept of time and space. Moreover, the neurosymbolic architecture is constructed for a fixed complex event specification, with a new complex event requiring a new model to be constructed from scratch. The project aims to make research advances that address these limitations. While the DAIS-ITA research on neurosymbolic methods for complex event processing was groundbreaking [Vilamala20, Vilamala21, Xing19, Xing20], the broader concept of integrating neural and symbolic approaches has previously been studied in AI. A first wave of research in 1990s sought to convert symbolic models into neural networks for further fine-tuning with data [Towell90], extract symbolic programs from neural networks learnt from data for further fine-tuning by domain experts [Towell93], and combine the preceding two to go back and forth [Shavlik94]. These early attempts at neurosymbolic architectures however did not have an impact as neural networks were not yet sufficiently performant for real-world usage due to lack of training data and hardware accelerators. In recent years, with real-world deployments of DNNs leading to concerns about their limitations [Bengio19], there has been a revival of interest in neurosymbolic approaches [Besold17, Chaudhuri21]. In the field of program synthesis there has been the emergence of neurosymbolic programming and its applications in symbolic regression for scientific discovery [Cranmer20], assistive tools for software developers [Ellis17], control of autonomous systems [Xu18], etc. It has the goal of synthesizing a program with both symbolic and neural components from a high-level task specification while simultaneously meeting hard logical constraints, approximately fitting a dataset, and generalizing to novel inputs. The recent research in neurosymbolic programming has studied both neurosymbolic learning algorithms and neurosymbolic program representations, providing useful insights such as the use of metalearning to train neural networks that generate the high-level program architecture [Balog16], and methods to distill neural networks into symbolic programs [Verma18] and to relax symbolic programs into neural networks [Verma19, Cui21]. Another relevant strand of recent research on neurosymbolic systems is represented by [Valkov18, Murali19, Cingillioglu21, Cunnington21] which all operate on images and process them with neural networks followed by symbolic logic to perform tasks such as visual discrimination and reasoning over objects in the image. While relevant, neither the research in neurosymbolic program synthesis nor the research in visual reasoning over images, address challenges in neurosymbolic complex event processing for situational awareness we target. It requires neurosymbolic architectures that process sensory data streams, incorporate spatiotemporal information, account for uncertainty, meet real-time requirements, be resilient to adversarial actions, and adapt to changing knowledge. Besides our prior work, research such as [Apriceno22] and [Niecksch23] have recently also explored complex events over unstructured sensory data from theoretical and experimental systems perspective.

NEUROPLEX++ DETECTION AND RECOGNITION ENGINE

Development and refinement of the detection and recognition engine called for incorporation domain expertise and data-driven learning, as well as the need to demonstrate classification performance across the CEP solution. The key challenge was ensuring that the components that constitute the *perception* layer of the architecture and process the raw, unstructured, high-dimensional sensory data into AEs are appropriately tuned for military relevant AEs. The existing NeuroPlex system [Vilamala21] targeted CEs in civilian settings with relatively simple AEs that only required object classification in a

video frame, acoustic event classification in sound, and human motion activity in inertial data, all of which it was able to perform using off-the-shelf pre-trained DNN models for object or event classification with reasonably high accuracy. However, the existing version of NeuroPlex did not include native capabilities for detecting and classifying military-relevant AEs in scenes with multiple dynamic objects. AE detection in such scenes requires detecting, classifying, and localizing objects in video frames; then tracking them over time across frames; and performing spatiotemporal reasoning across time, space, and multiple sensor data streams. The existing system lacked these capabilities and had to be enhanced.

The required capabilities were introduced by re-engineering the neurosymbolic pipeline to expand its capabilities as well as devising a language for specifying the AEs and CEs to handle necessary spatial concepts such as trip wires, watch boxes, etc. The neurosymbolic pipeline consists of two stages: a single-pass object detection DNN for each sensor which outputs a set of objects, including their type and bounding boxes [Wang22], and a two-part symbolic processing stage. The first part of symbolic processing performs tracking and reidentification of objects across video frames while undertaking rule-based measures to mitigate detection and tracking errors, and the second part detects the AEs and the CEs using finite state machines (FSMs) generated from the AE/CE specification language.

The neurosymbolic pipeline thus expands roles for both the neural and the symbolic stages relative to its predecessors: the neural processing now performs detection and localization instead of just classification, and the symbolic processing now performs tracking, reidentification, run-time error mitigation, and spatiotemporal reasoning relating to AE/CE detection instead of only detecting temporal patterns that constitute CEs.

Incorporating multiple objects, spatial reasoning, and temporal patterns. While evaluation of complex spatiotemporal events is typically more effectively done with first-principal models that are maximally generalizable, analysis of high-dimensional unstructured sensor data is typically better approached by data-driven models. The neurosymbolic approaches represented by DeepProbCEP [Vilamala23] and the original NeuroPlex provide a balanced combination of the advantages of these approaches. Unfortunately, first generation neurosymbolic CE approaches targeted settings that required only object classification (no detection, localization, tracking, etc.) and detection of temporal patterns over sequences of classification labels (no spatial reasoning). This required an improvement beyond the use of neural proxies interpreting symbolic information into exact arithmetic circuits to backpropagate new information about CE prediction into the trained object and event classifier(s) [Gan18].

These needs were addressed by the introduction of an *object detector* based on YOLOv5, a single-pass DNN based model, discussed in greater detail below [Zhu21]. This served as the basis for development of an *object tracker*, which re-identifies and tracks objects across frames, and mitigates errors and confounders in object detection. Two symbolic units were designed for event detection:

- **AE Detector:** Detects spatial patterns of objects over short time intervals using abstractions of *watch boxes* and *trip wires*.
- **CE Detector:** Detects temporal patterns of AEs occurring asynchronously and irregularly over long spans of time.

The CE specification language was expanded to accommodate multiple objects per sensor sample, incorporate object locations, multiple sensors, and spatial abstractions, and allow declarative expression of temporal patterns. This modified language was used as the basis for analysis of the

performance of the CE detection performance under various conditions. Functionality of the system is outlined below in Figure 2.

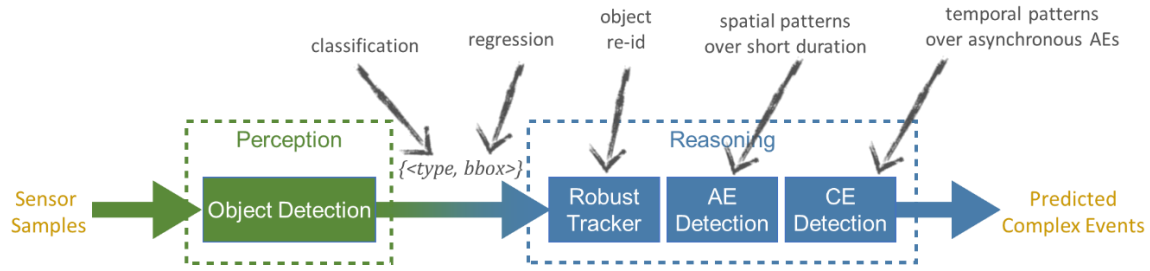

Figure 2. Implementation of neurosymbolic reasoning.

Object Detection. The YoloV5 model we used for object detection was trained as follows: We used an off the shelf, state of the art detector with a well-supported software base [Zhu21]. This tool was trained on 8,800 images across 5 object classes, including a T-72 tank with a neutral color scheme, a BTR reconnaissance vehicle, and three different classes of civilian vehicle. Training was started with a network of medium size, using a v6 model. The outputs yielded by the detector included a bounding box for each targeted entity, object confidence, and SoftMax confidence estimate for each class. This capability informed the phase II plan for object tracking using Kalman Filters. Figure 3 is an illustration of the output of the Object Detector performance, including entity class detected and confidence level associated with the classification.

Figure 3. YoloV5 Object Detector Class and SoftMax Confidence Level Output

Generalizing performance to operational conditions for which the system was not specifically trained. When conditions change, the existing library of entities, AE, and CEs a CEP system has been trained to recognize may no longer be sufficient. Domain shifts can take the form of new environments or conditions under which a TTP is exercised or attempted, they may involve the modification or adoption of entirely new TTPs or unit-level behaviors. They may simply involve changes in the reliability, availability, or periodicity of sensor data upon which event recognition was based. Any such

changes may mean that the CEP system will need to adapt or be modified in order to work in this new domain – whether by modification of detection criteria, specifications of new entities or events, or environmental changes that force redefinition of AE/CEs in the existing library.

We argue that domain shifts to be accommodated will fall into three broad categories, *perceptual*, *behavioral,* and *environmental* shifts. If the causes of uncertainty triggers appear to be driven by the inability of the sensor(s) to detect behaviors that are relatively unchanged, the task for the SME may be to define or revise existing detection criteria, accommodating a *perceptual domain shift*. Perceptual domain shifts will call into question whether the AEs and CEs targeted by the system's existing models will still be detected based on reduced signal quality. For example, if a sensor is occluded by weather conditions so that the sensor can only detect 50% of the behaviors indicating a red force attack is imminent, should the system infer an attack likely if 40% of the entity behaviors expected are observed by the sensor? The SME will make this determination and input guidance into NP++ using the CEP grammar UX. Conversely, a *behavioral domain shift* will involve the definition of new entities, behaviors, potential modifications of AE/CE, and respecifications of the rules governing interaction of entities with each other and their environments. Environmental domain shifts can be defined as changes in the terrain or environment in which the sensors are deployed causing degraded performance for the neural components. The impact of terrain or environmental changes can be conceptualized using a combination of perceptual or behavioral effects, given their potential impacts on sensor performance and how, where, and when entities move.

SYNTHETIC DATA GENERATED FOR PERFORMANCE EVALUATION

Development of Scenarios and TTP Variations. The team developed systematic, scalable, labeled scenario files containing relevant, accurate, varied iterations of background, AE, and CE behavior in appropriate environments by red and background POL entities for training and evaluation of the NeuroPlex++ system.

Relevance of Use Case. The first challenge was development and simulation of an appropriate use case that serves as a tactically relevant, plausible instance of CEs underlying adversary TTP demonstration across operational conditions. The use case also needed to provide an operational example of multiple CEs and underlying AEs exemplifying the target TTP(s), in the presence of background entities. The scenario data representing this use case had to be implemented as an accurate representation of entities executing the AEs and CEs. These behaviors had to be modeled appropriately as raw sensor data for ingestion by NeuroPlex++.

Scenario Modeling. Scenarios needed to be designed that would provide demonstrations of tactically relevant atomic and complex events that could be *detected* by simulated sensors [Xing21]. The AEs and CEs needed to be implemented in an appropriately modeled environment, that afforded the right level of fidelity for environmental features and details constraining entity movement. The environment needed to provide a realistic instance of the level of granularity with which roads, features, and entities could be described while maintaining a reasonable burden of processing time for generation and rendering. The acuity, zoom level, and altitude of the simulated airborne sensors to be taken into account as well, to ensure that the entities targeted for recognition were depicted using an appropriate number of pixels so as to provide a meaningful challenge for a trained object detector [Sun21].

Data Synthesis. The tools and processes to be used for synthesis of the environments, entities, and sensor feeds needed to be defined such that the scenarios in which AEs and CEs were modeled could be generated at scale, with appropriate entity behaviors, goals, waypoints, and sources of variation across iterations defined for both red and background entities [Stensrud12]. The scenario files generated also needed to include a usable number of negative examples in which AEs and CEs did not occur for purposes of training. Finally, the data needed to be labeled appropriately with ground truth for entity positions and event occurrence.

As depicted below in Figure 4, the team developed a series of scenarios for Phase I taken from real world conflict use cases in Eastern Europe. As was widely depicted in global news in 2022, the problem of making predictions about red forces moving armor/tank columns in and around contested bridges was of significant tactical relevance to the War in Ukraine. Elements of these real-world situations were adapted for the phase I use case, simulating a defense by blue forces of a series of bridges in the Ann Arbor MI metropolitan area. The original scenario included the decisions regarding blue force deployment in response to predictions of enemy intentions when moving through three different named areas of interest (NAI).

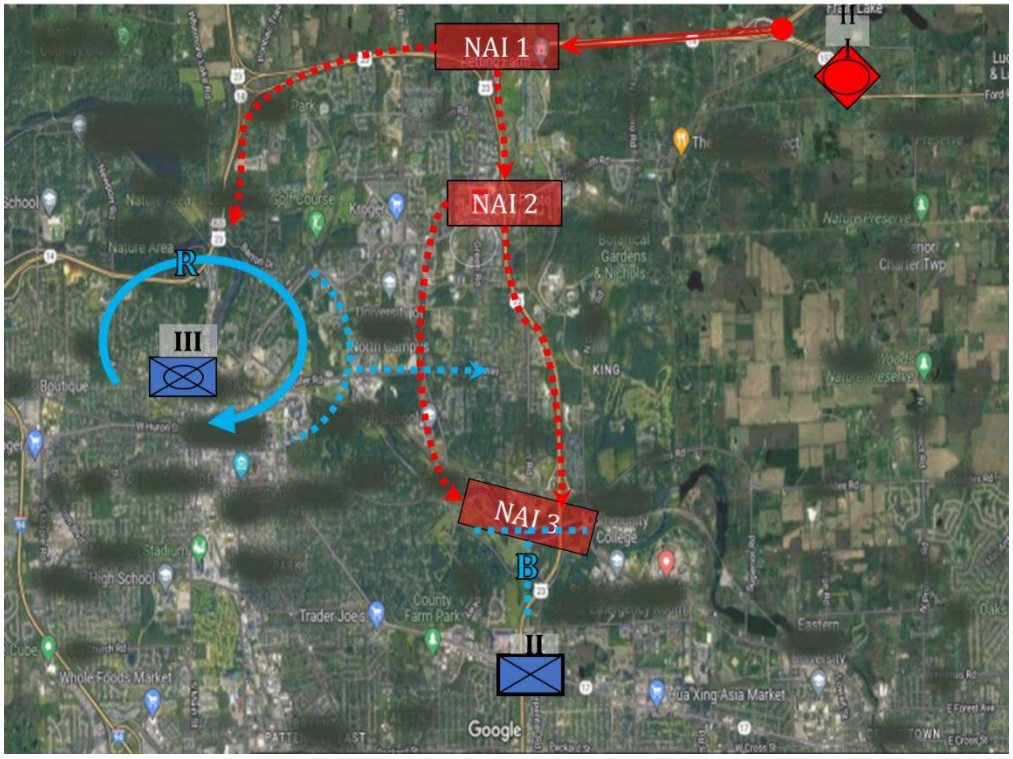

Figure 4. Tank column bridge scenario model

Conditions of the scenario were as follows:

- Orientation: Enemy armored forces are moving west along Highway 14 with the intent of crossing the Enborne river.
- Situation: The Regiment is established West of Highway 23 in a blocking position south of the Enborne River.

- Mission: On order Battalion forces establish a blocking position south of the Enborne River (in view of Hwy 23 bridge) in order to drive enemy armored forces to a western river crossing (in view of downtown).
- Execution: The Regiment has emplaced obstacles at the intersection of Ply Rd and Highway 23 (NAI 2) with the intent of pushing the enemy toward the western river crossing along Huron Pkwy. The Battalion will maintain its blocking position south of the river while the Regiment engages the enemy along Huron Pkwy.
- Admin and Logistics: All re-supply and logistics requirements are handled at the Regimental level.
- Command and Signal: Battalion retains OPCON of all organic fire support, aviation, and ISR assets.

A limited subset of this scenario was implemented as the initial AE/CE recognition challenge described here, using three simulated fixed position aerial surveillance assets between NAIs 2 and 3.

Development of Complex Event Taxonomy. Behavior of the entities used to define the AEs and CEs depicted next was organized using a taxonomy developed for this effort, which began with the detection and tracking of objects or entities as the base level recognition task. Sensors were modeled as electrooptical cameras that recognized presence of entities for which they were trained at the individual frame level. These data were not used to emulate sensor tracks for purposes of this effort. Detected objects were used as the basis for defining AEs, which were triggered by watchbox or tripwire crossing by targeted entities with different relative positions or separation from other specific entities (i.e., distance from the other tanks in a column). AEs were used here to capture the location and presence of single or multi-entity groups over short spatio-temporal distances. AEs were aggregated into CEs, which required the stitching together of information from multiple sensors, over greater periods of spatio-temporal distance. CEs were designed to capture change over time, and are defined as the most narrowly specified multi-entity events that could be expected to have tactical relevance to an intel analyst witnessing them.

In other words, a CE is one that an attentive analyst would be expected to recognize and attribute tactical implications of a set of disparate movements by multiple entities across time and sensors. CEs are the highest order events modeled in the Phase I data, but they are designed to be consolidated into military tactics, techniques and procedures (TTPs).

Development of AE/CE Scenarios. Using this taxonomy, and a limited subsection of the original scenario, we implemented the following set of Complex Events and underlying atomic events. Using a portion of the Ann Arbor map spanning NAIs 2 and 3, we modeled three fixed-position aerial sensors at a simulated altitude of 400’ above ground level, sufficient to render objects for which our YoloV5 object detector at a number of pixels needed to achieve plausible performance levels [Martinson21]. If the simulated altitude had been raised much higher, we would have been able to observe many more entities simultaneously in each sensor’s field of view, but object detection would have suffered as a result, likely rendering the detection of all events impossible.

The sensor watchboxes rendered and specific terrain captured in the fields of view of each simulated sensor are depicted in Figure 5 (See Appendix B for more detail on these modeled sensor positions). Sensor 1 was modeled as two concentric watchboxes directly north of Highway 23 of dimensions 500m x 280m and 350m x 100m. Sensor 2 was modeled using the same structure at a position 2000m north of the Enborne River, and Sensor 3 was modeled as a viewing area of the same size, with watchboxes

defined on the north and south sides of the river used to monitor for reconnaissance vehicles in advance positions.

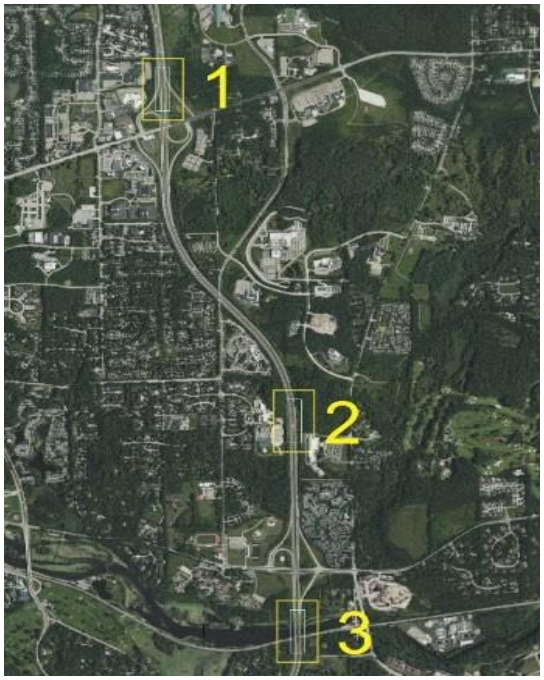

Figure 5. Fixed sensor positions

We defined three different CEs involving movement of a column of tanks and recce vehicles through Ann Arbor toward a bridge. In all CEs, the column consisted of 4-6 tanks moving in two staggered columns with targeted separation of 20m between the nearest neighbor when in formation. The column was escorted by a set of 4 reconnaissance vehicles that moved in pairs, with targeted separation from the tank column of two minutes, and a targeted distance between the two pairs of reconnaissance vehicles of 50m during travel. All roads were also populated by a randomly distributed set of 400-500 civilian vehicles. Below is a summary description of each complex event. Exact details are provided in Appendix A.

- CE1 Column preparing to detonate the bridge, as indicated by reconnaissance vehicles taking positions on both sides of a bridge in pairs, in the road, blocking civilian traffic, at a distance of >100m from the ends of the bridge, with the tank column not moving toward the bridge, and the far pair of reconnaissance vehicles rejoining the column after two minutes in position.
- CE2 Column preparing to cross the bridge, as indicated by reconnaissance vehicles taking positions on both sides of a bridge in pairs, off road, not blocking civilian traffic, with the tank column moving toward the bridge.
- CE3 Column taking up a defensive position, as indicated by reconnaissance vehicles moving in pairs in all directions away from a planned defensive position and stopping to monitor traffic/threats from fixed positions away from the tanks. Tank column breaks formation and forms into a line or an arc between the sets of reconnaissance vehicles.

Each scenario “take” consisted of video feeds representing the same 10 minute period captured by 3 stationary airborne sensors. Red entities consisted of a column of 4-6 tanks escorted by a set of 4

recce vehicles. POL entities consisted of civilian auto traffic. Occurrence of all these AEs took place over a period of 5-10 minutes and involved detection of objects monitored by 3 different sensors.

Development of the Unity Simulation Environment, NP Sim. The system used to generate the scenario data described above was a Unity-based capability called NP Sim. This tool is capable of defining the entities to be modeled in a scenario, along with their goals and behaviors managed by a scenario controller as depicted below in Figure 6. More importantly, it automates the process of scenario generation, execution, and recording, enabling the generation of an unlimited number of variable iterations of scenarios involving targeted behaviors, and boundaries for their variation across scenario iterations.

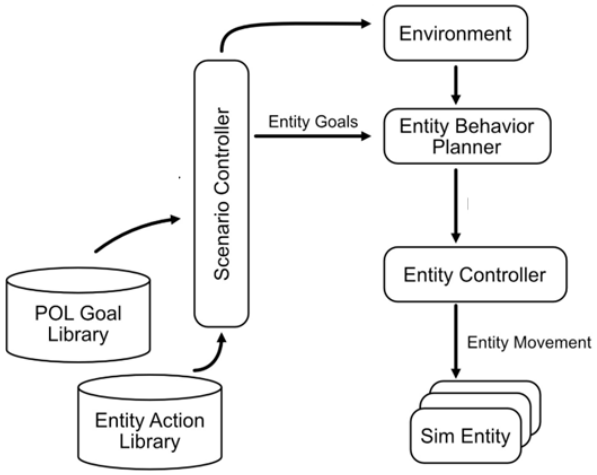

Figure 6. Architecture of the NP Sim tool.

The behavior of the red entities was operationalized as a set of emergent actions based on a waypoint following behavior, an obstacle avoidance behavior,[Heilbron15] based on assigned destination targets, interactions with each other and POL vehicles, and starting locations on the map. Refined movements at critical points for increased probability of AE/CE engagement were defined using clusters of waypoints. Red entity routes were governed by a low density directed graph of waypoints, selected using in the entity controller as part of a state machine defined as part of the scenario. The selected entity goals directed the order in which targeted actions, including all AEs and CEs, were attempted within the scenario.

In functional terms, the most important parameter used to manage red entity formation adherence was entity speed. Tanks and reconnaissance vehicles were assigned grouping characteristics, with each group assigned a random speed within a specified range. Individual vehicles within groups had interval/separation targets they attempted to adhere to by varying their speed, and direction insofar as directional variation was necessary to avoid collisions.

This process also supported the generation of more sophisticated POL entity traffic behaviors, ensuring comparability between POL/background AEs and red entity AEs. POL and red behavioral comparability in the synthetic data was necessary to provide a meaningful recognition challenge for the tool. POL entities utilized a *high-density waypoint following* system built along roads via COTS traffic asset for Unity, the Gley Traffic System.

POL vehicles started at random positions on the roads anywhere on the modeled terrain. Red Entity starting points were variable, but all north of sensor 1 in this data corpus. Note that additional variation was introduced by the emergent behaviors that resulted from the interaction of the behavior controller and the Unity physics system. This introduced variation across iterations due to this interaction regardless of whether the scenario controller, the entity planner, and the entity controller all dictate identical constraints and direction to the entities modeled within a scenario. So while seed values dictating how an NPSim scenario can be held static between takes, there are no full scenario iterations that can ever be completely deterministic due to the interaction of these entities with each other [Vaswani17]. The non-deterministic nature of these scenario iterations is desirable due to the contribution it makes to execution variability (within constraints) across scenarios, but it does mean that there can be iterations where targeted behaviors and goals defined by the scenario controller and entity controller are not successfully executed [Heilbron15, Herath17]. The incidence rate of failure to execute assigned behaviors across iterations appears to be extremely low, which could be verified using the ground truth data also produced by NPSim.

Data Labeling. NPSim ground truth information is captured in a log file generated for every scenario iteration. This file captures when the state machine transitions between states. The process by which entity behavior is modeled across iterations includes explicit, translatable criteria for instances where mapped AE and CE events occur and capturing entity positions frame by frame. This readily yields geographic coordinates within a sensor box for location of all AEs, CEs, and targeted entities (in phase I, limited to red entities) along with a timestamp at the frame level for the metadata produced.

Data Corpus Generated. The list below in Table 2 summarizes the 311 scenario variations generated to model each of the complex events described above, as well as incomplete CE demonstrations or scenario takes in which no CE was modeled. Metrics used for comparison and for exploratory purposes are discussed next, along with results of all detection and false alarm analyses conducted on original environment and domain shifted scenarios. Two varieties of perceptual domain shift were implemented as a representation of our goal to generalize NeuroPlex++ performance to different operational environments:

a. Intermittent wildfire smoke introduced as an occlusion that blocks sensor visibility of entities
b. Variation of hostile entity appearance: We introduced variations of the T-72 tank design for which YoloV5 Object Detector (OD) was not trained

Multiple variations of smoke opacity and tank design were introduced to ensure that the domain shift change provided an appropriate challenge to the YoloV5 OD. Initial versions of each resulted in zero object recognition in the wildfire smoke example, and 100% object recognition using the first version of the T-72 tank variation, neither of which were useful levels of baseline performance for assessing functionality of the symbolic detection algorithms designed for AE and CE detection. Examples of both types of perceptual domain shift introduced are illustrated below in Figure 7.

CE	Takes	Variations
1	**38 complete**	118 variations: **37 incomplete** **54 DS alternate tank** **27 DS smoke**
2	**41 complete**	3 DS alt tank 0 DS smoke
3	**52 complete**	2 DS alt tank 1 DS smoke
None	**49 no AEs**	3 DS alt tank 4 DS smoke

Table 2. Data corpus by Complex Event
Note: alt = alternate; DS = domain shifted.
Scenarios counts in boldface are reported in subsequent analyses

Figure 7. Examples of perceptual domain shift introduced in scenarios

While the work reported here is limited to demonstration that CEP detection could be maintained under domain shift, a far more important question for continuation of this work concerns how system recognition of domain shift could be defined and triggered. As a result of our data synthesis and architecture updates, the team was able to specify a needed mechanism for recognition of conditions under which generalization to other environments or recognition of new tactics was needed.

Pending improvements will associate quantitative *uncertainty* measures for neurally derived object detection. Multiple techniques for one-stage object detection have already been discussed, including the approach introduced by He [He19] that focused on uncertainty in bounding box regression, and that focused simultaneously on two sources of variance: bounding box transformation and localization. Other researchers [Kraus19] have looked at dropout as an approximation technique for prediction distribution of a Bayesian NN, while Lyu [Lyu16] combined deep ensembles and Monte Carlo dropout for uncertainty estimation. These one-stage uncertainty estimates will be assessed first as candidates for definition of domain change thresholds. Failing this, the viability of multi-object tracking algorithm uncertainty indices will be investigated. These uncertainty estimates will be used to

define thresholds indicating to the system that a domain change has occurred, and that SME updates to the entity and event library using the CEP Grammar UX are urgently needed. Those SME updates should make it possible to accommodate all types of domain shift, while maintaining CE recognition accuracy with a minimal set of SME-defined training examples.

ASSESSMENT OF NEUROPLEX++ PERFORMANCE AT COMPLEX EVENT DETECTION

The solution described relied on neurally trained tools for detection of relevant objects, and symbolic reasoning for the detection of AEs and CEs. As stated above, the improvements to the first generation neurosymbolic CE approaches previously developed under DAIS ITA were not suited to detection, localization, or tracking, and were not capable of spatial reasoning; they had previously been used for classification and temporal reasoning only. The improvements we introduced to address this need includes neurally based object detection using the single-pass YOLOv5 DNN model. The updated system also includes a robust tracker that reidentifies and tracks objects by comparing the changes in the relative positions of object IDs across adjacent frames and mitigates errors and confounders in object detection.

We also redesigned the CE specification language to support multiple objects per sensor sample, object locations, multiple sensors, and spatial abstractions, and allow declarative expression of temporal patterns. We used this to analyze and improve CE detection performance under various conditions. Event detection was managed by two symbolic units designed for this purpose: an AE Detector that detects spatial patterns of objects over short time intervals using abstractions of *watch boxes* and *trip wires,* and a CE Detector that captures temporal patterns of AEs that can occur in many different orders and with different temporal separation over long spans of time. Event detection accuracy rates and false alarm rates were measured according to the operational definitions below.

1. Accuracy: The proportion of takes for which the identified CE and underlying AE set matches or fails to match ground truth information[1].
2. Missed detections: number of AEs or CEs which were missed but occurred in the ground truth.
    - Complex Event detection accuracy and missed detection count are reported for the 38, 41, and 52 takes of complete occurrence of the AEs contributing to CEs 1, 2, and 3, respectively, in Table 3.
    - Complex Event detection accuracy and missed detection count *under domain shifted conditions* are reported for the 54 CE1 takes involving wildfire smoke and the 27 CE1 takes involving alternative tank coloring schemes on which the OD was not trained.
3. False alarms: number of AEs or CEs which the NeuroPlex++ system detected but did not occur in the NPSim ground truth logs.
    - False alarm analyses were only conducted on the takes with incomplete execution of CE contents (37 takes involved subsets of the AEs contributing to CE1) or those with zero intentionally included AEs (the 49 takes listed in Table 3)

[1] It is worth clarifying here that in this specific set of scenarios, the CEs we defined required the detection of all component AEs, but in other settings and circumstances, this may not be the case.

Metric thresholds for detection accuracy were defined using a synthetic dataset representing three different CEs, each composed of at least three different AEs, with AE elements separated by space and time, and captured by multiple sensors. Thresholds for detection accuracy for each CE was **90%** in the initial operational environment modeled, with a false alarm rate of **no more than one false alarm per hour** (assuming some constant rate of evaluation over time). The CEs were then to be modeled in a second operational environment, and the CE detector performance was evaluated in this second environment after being provided minimal training data. Detection accuracy threshold in the second operational environment was **80%**.

Summary results are provided below in Table 3. The team achieved detection accuracy targets for all 3 implemented CEs, across a total of 131 three-sensor scenario executions, as well as another 37 takes in which a subset of the AEs making up CE 1 were incorporated. False alarm rates across 86 takes comprised of: an incomplete set of AEs underlying CE 1 (N = 37); or takes in which no red entity AEs were included (N = 49) were 1 and 0, respectively. ***In the initial operational environment, all performance thresholds were met.***

The additional operational environments used to test the performance of NeuroPlex++ for environments in which it had not been explicitly trained included 54 takes of CE 1 execution by a variation of the T-72 tank with a different paint scheme on which the YOLOv5 DNN had not been trained and 27 takes of CE 1 in which smoke from a stationary wildfire periodically obscured visibility of ground entities to the sensors. Execution of AEs underlying CE1 were accurately detected in 45 of 54 cases involving the alternate T-72 paint scheme, for an accuracy rate of 83.3%. In the case of the wildfire smoke, all AEs underlying CE1 were detected in 18 of 27 cases, for a detection accuracy rate of 66.7%. ***In the modified operational environments, therefore, detection accuracy was mixed, with the 80% threshold achieved in one environment, but not the other.***

Scenario Event	Accuracy (Count)			FA Rates
	Base Scenario	DS 1 alt tank	DS 2 smoke	N/A
CE 1 Bridge Detonation	97.3% (37/38)	83.3% (45/54)	66.7% (18/27)	N/A
CE 2 Bridge Crossing	92.6% (38/41)	insuff. data	insuff. data	N/A
CE 3 Defensive Position	94.3% (49/52)	insuff. data	insuff. data	N/A
Incomplete CE 1	97% (36/37)			1/37
No CEs	N/A			0/49

Table 3. Summary accuracy and false alarm rates.

Note: CE = Complex Event, DS = Domain Shift, FA = False Alarm, insuff = insufficient

Complete CE1 Takes N = 38. Out of a corpus of 38 complete instances of CE1, bridge detonation, we successfully detected CE1 in all instances except one, yielding CE **detection accuracy of 97.3%**. The missed AE was a single instance of 1.1.c (as defined in Appendix A), which involves reconnaissance

vehicles on the far side of the bridge leaving their posts after blocking traffic to rejoin the red entity tank column.

Incomplete CE1 Takes N = 37. Out of a corpus of 37 instances of incomplete execution of CE1, we successfully detected all the relevant AEs in 36 of those cases, for a completed event detection accuracy of 97%. In these takes, the recce vehicles completed requirements for the first three AEs, but did not leave their positions at the south side of sensor area 3 to rejoin the tank column following traffic blocking in 1.1.b. It is also worth noting that in 7 of these 37 instances, NeuroPlex analyses identified an error in the NPSim ground truth log. In the set of 37 takes in which an incomplete set of CE 1 AEs occurred, there were 6 takes in which the ground truth data yielded by NPSim failed to recognize the occurrence of AE 1.1.b., in which a formation of reconnaissance vehicles take position on both sides of a bridge, obstructing the road at a distance of >100m from the ends of the bridge, so that POL traffic is blocked from entering the bridge. There was only one false alarm at the CE level: 1/37 (2.7%).

Complete CE2 Takes N = 41. Out of a corpus of 41 instances of execution of all AEs underlying CE2, bridge crossing, we achieved event detection accuracy of 92.6%. There were three instances of failure to detect AE 2.1c, movement of the tank column toward the bridge.

Complete CE3 Takes N = 52. Out of a corpus of 52 instances of execution of the 3 AEs underlying CE 3, defensive formation, we have completed event detection accuracy of 94.3%. There were three instances in which AE 3.1, reconnaissance vehicles take up positions on outbound roads away from the tank column, was not recognized due to proximity of POL entities to reconnaissance vehicles, the positions of those reconnaissance vehicles relative to each other once they took up their observation posts.

No atomic or complex events Takes N = 49. There were zero false alarms on these takes.

Complete CE 1 Takes with Domain Shift: Alternative T-72 Tank N = 54. We generated a corpus of 54 instances of CE 1 execution by a group of tanks whose external coloring was a different camouflage pattern than the grey scheme upon which the YOLOv5 object detector was trained. Of these 54, we achieved event detection accuracy of 83.3% The missed AEs were nine instances of 1.1.c, which involves reconnaissance vehicles on the far side of the bridge leaving their posts after blocking traffic to rejoin the red entity tank column. In five of those nine instances, the OD failed to recognize the reconnaissance vehicles when they took flanking positions on opposite sides of the bridge to block POL traffic.

Complete CE 1 Takes with Domain Shift: Wildfire Smoke N = 27. We generated a corpus of 27 instances of CE 1 execution while a fixed position wildfire generated smoke of limited opacity with wind pattern shifts cycling every 30 seconds, partially obscuring the sensor window's exposure to targeted entities, and impacting the object detector's ability to recognize the targeted entities. As was the case in the first domain shift example presented above, performance suffered most at detection of the two AEs involving reconnaissance vehicles taking position on opposite sides of the bridge to block traffic (1.1.b) and at detection of the pair of reconnaissance vehicles on the far side of the bridge departing their post to rejoin the tank column (AE 1.1.c). Failure to detect this AE 1.1.c. departure coincided with the 6 failed instances of detection of the reconnaissance vehicles taking position on opposite sides of the

bridge per AE 1.1.b, for a total of 9 missed observations out of 27, or a completed event detection accuracy of 66.67% for the wildfire-introduced domain shift.

In summary, results suggested that the re-engineered neurosymbolic pipeline augmented with expanded event processing language was an effective mechanism for detection of atomic and complex events in simulated military domain operations data. Recognition threshold targets were met for all CEs in the original operational environment, and false alarm rates were at or below acceptable thresholds. In the two separate instances of perceptual domain shift evaluated, recognition threshold results were mixed, with targeted recognition above 80% achieved in the alternative tank domain, but recognition at 67% in the domain shift instance involving a stationary wildfire occasionally obscuring line of sight between sensors and targeted entities.

DISCUSSION

Derivation of meaningful capability in this domain depends on several critical enablers. Developers must be equipped to define military domain problems according to a hierarchy of complexity, consistent with a workable entity and event taxonomy. Work with experts under this effort provided insight into the optimal level of detail and mechanism to capture from subject matter experts, including the potential utility of capturing SME input using a combination of statements and a graphical user interface.

It is critical to have an extensible architecture for generation of new scenario data featuring greater control and modification of terrain and structure elements, entity type and entity count, and a richer library of entity behavioral controls and goals. Mechanisms and alternatives for the generation and validation of ground truth data at the entity and event level were also researched.

Generation of a specification language for capturing entity and unit behaviors and characteristics, defining events, and capturing SME input is immensely useful. This extended to mechanisms for working with multiple sensors, incorporating spatial abstraction, and description of temporal patterns. Mechanisms for object detection and tracking were also devised and improved.

Finally, generalizability and deployability of such a solution depends heavily on understanding the appropriate and needed differences between baseline and domain-shifted data via testing and iteration. Understanding what to change and to what degree in order to model a perceptual or behavioral domain shift is critically valuable knowledge. This insight is important for ensuring that an appropriate reduction in baseline entity/event perception relative to an “unshifted” scenario modeled in the original operational environment is defined, in order to present a meaningful challenge to the CEP tool.

CONCLUSION

The NeuroPlex++ effort achieved virtually all its performance targets. More importantly, the work performed and innovations generated represent a highly promising avenue for further innovation. The team was able to generate insightful, relevant, and organized red entity scenarios and associated behaviors, defined in the bounds of a usable object and event taxonomy traceable up to TTP content.

The Unity-based NPSim tool is capable of generating high-count variable iterations of combined urban scenarios featuring plausible, variable goal-oriented behaviors by background and red entities, including constraints on how entities interact with each other. The system also automatically captures ground truth for event occurrence and entity position frame-by-frame.

The existing NeuroPlex tool was improved to accommodate viewing multiple objects per sensor sample, incorporate object locations, multiple sensors, and spatial abstractions, and to allow declarative expression of temporal patterns. The inclusion of an object detector and object tracker in NeuroPlex++ allowed the tool to expand beyond its previous limits of classification, and a mechanism for the introduction of perceptual domain shift.

Finally, the team developed a mechanism for the OD-based recognition that a domain shift has taken place, which will serve as a critical enabler of future innovations. These include further maturation of the NPSim environment and capability, enhancements to the NeuroPlex++ system itself by direct incorporation of transformer models for neurosymbolic recognition of AEs, and improved self-training and neurally reconstructed logic [Manhaeve18]. The plan also calls for adoption of one of a series of promising OD uncertainty estimations to be used as the basis for determining when domain shift has occurred, necessitating SME updates, will expand the neurosymbolic CEP grammar, and deliver a UX by which SME expertise will be more readily captured and catalogued.

AVAILABILITY OF DATA FOR RESEARCH PURPOSES:

Interested researchers are advised that the data files and YOLOv5 model trained for this effort are available for download by interested researchers. Visit the following address to initiate a request for access to the data: https://github.com/nesl/ComplexEventDatasets

ACKNOWLEDGEMENT

The research in this paper was sponsored by the DEVCOM Army Research Laboratory through STTR contract W911NF22P0064. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government."

REFERENCES

[Ahmed22] Ahmed, K., Li, T., Ton, T., Guo, Q., Chang, K.-W., Kordjamshidi, P., . . . Singh, S. (2022). Pylon: A PyTorch framework for learning with constraints. Association for the Advancement of Artificial Intelligece.

[Alevizos17] Alevizos, E., Anastasios, S., Alexander, A., & Paliouras, G. (2017). Probabilistic complex event recognition: A survey. ACM Computing Surveys (CSUR), 50(5), 1-31.

[Apriceno22] Apriceno, Gianluca, Andrea Passerini, and Luciano Serafini. "A neuro-symbolic approach for real-world event recognition from weak supervision." In *29th International Symposium on Temporal Representation and Reasoning (TIME 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[Balog16] Balog, Matej, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, and Daniel Tarlow. "Deepcoder: Learning to write programs." arXiv preprint arXiv:1611.01989 (2016).

[Bengio19] Bengio, Yoshua. "From system 1 deep learning to system 2 deep learning." In Thirty-third Conference on Neural Information Processing Systems. 2019.

[Besold17] Besold, Tarek R., Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger et al. "Neural-symbolic learning and reasoning: A survey and interpretation." arXiv preprint arXiv:1711.03902 (2017).

[Cakir17] Cakır, Emre, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. "Convolutional recurrent neural networks for polyphonic sound event detection." IEEE/ACM Transactions on Audio, Speech, and Language Processing 25, no. 6 (2017): 1291-1303.

[Chaudhuri21] Chaudhuri, Swarat, Kevin Ellis, Oleksandr Polozov, Rishabh Singh, Armando Solar-Lezama, and Yisong Yue. "Neurosymbolic Programming." Foundations and Trends® in Programming Languages 7, no. 3 (2021): 158-243.

[Cingillioglu21] Cingillioglu, Nuri, and Alessandra Russo. "pix2rule: End-to-end Neuro-symbolic Rule Learning." arXiv preprint arXiv:2106.07487 (2021).

[Craig21] Craig, R., Mercado, J., Kawatsu, C., and Purman, B (2021), "Computer Vision Aided Unexploded Ordnance (UXO) Detection using Synthetic Data," in *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, Orlando, FL

[Cranmer20] Cranmer, Miles, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. "Discovering symbolic models from deep learning with inductive biases." arXiv preprint arXiv:2006.11287 (2020).

[Cui21] Cui, Guofeng, and He Zhu. "Differentiable Synthesis of Program Architectures." Advances in Neural Information Processing Systems 34 (2021).

[Cunnington21] Cunnington, Daniel, Mark Law, Alessandra Russo, Jorge Lobo, and Lance Kaplan. "Towards Neural-Symbolic Learning to support Human-Agent Operations." In 2021 IEEE 24th International Conference on Information Fusion (FUSION), pp. 1-8. IEEE, 2021.

[Ellis17] Ellis, Kevin, Daniel Ritchie, Armando Solar-Lezama, and Joshua B. Tenenbaum. "Learning to infer graphics programs from hand-drawn images." arXiv preprint arXiv:1707.09627 (2017).

[Gan18] Gan, Y. (2018). *Hardware acceleration for tensorized neural networks*. UC Santa Barbara Electronic Theses and Dissertations.

[Github, n.d.] YOLOv5. (n.d.). Retrieved from https://github.com/ultralytics/yolov5

[He19] Y. He, C. Zhu, W. Jianren, M. Savvides and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2888-2897*, 2019

[Heilbron15] Heilbron, F.; et. al.,, "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding," in CVPR, 2015.

[Herath17] S. Herath, M. Harandi and F. Porikli, "Going deeper into action recognition: A survey," Image and vision computing, vol. 60, pp. 4-21, 2017.

[Hinton15] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[Jeyakumar20] Jeyakumar, Jeya Vikranth, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. "How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods." Advances in Neural Information Processing Systems (2020).

[Jeyakumar23] Jeyakumar, Jeya Vikranth, Ankur Sarker, Luis Antonio Garcia, and Mani Srivastava. "X-CHAR: A Concept-based Explainable Complex Human Activity Recognition Model." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, no. 1 (2023): 1-28.

[Kraus19] F. Kraus and K. Dietmayer, "Uncertainty estimation in one-stage object detection," in *IEEE Intelligent Transportation Systems Conference (ITSC), pp 53-60*, 2019

[Law20] Law, Mark, Alessandra Russo, Elisa Bertino, Krysia Broda, and Jorge Lobo. "FastLAS: scalable inductive logic programming incorporating domain-specific optimisation criteria." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 03, pp. 2877-2885. 2020.

[LearnOpenCV, n.d.] Object Tracking using OpenCV. (n.d.). (Learn OpenCV) Retrieved from https://learnopencv.com/object-tracking-using-opencv-cpp-python/

[Luckham02] Luckham, D. (2002). The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Professional.

[Lundberg17] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." In Proceedings of the 31st international conference on neural information processing systems, pp. 4768-4777. 2017.

[Lyu21] Z. Lyu, N. B. Gutierrez and W. J. Beksi, "An uncertainty estimation framework for probabilistic object detection," in *IEEE 17th International Conference on Automation Science and Engineering (CASE), pp. 1441-1446*, 2021.

[Manhaeve18] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester and L. De Raedt, "DeepProbLog: Neural probabilistic logic programming," in arXiv: 1805.10872v2 [cs.AI] 12 Dec 2018.

[Martinson21] E. Martinson, B. Furlong and A. Gillies, "Training Rare Object Detection in Satellite Imagery With Synthetic GAN Images," in Proceedings of the IEEE/CVF Conf on Computer Vision and Pattern Recognition, 2021.

[Murali19] Murali, Adithya, and P. Madhusudan. "Augmenting Neural Nets with Symbolic Synthesis: Applications to Few-Shot Learning." arXiv preprint arXiv:1907.05878 (2019).

[Niecksch23] Niecksch, Lennart, Henning Deeken, and Thomas Wiemann. "Detecting spatio-temporal Relations by Combining a Semantic Map with a Stream Processing Engine." In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8224-8230. IEEE, 2023.

[Ribeiro16] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135-1144. 2016.

[Shavlik94] Shavlik, Jude W. "Combining symbolic and neural learning." Machine Learning 14, no. 3 (1994): 321-331.

[Singh16] Singh, Suriya, Chetan Arora, and C. V. Jawahar. "First person action recognition using deep learned descriptors." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2620-2628. 2016.

[Stensrud12] B. Stensrud, "No more zombies! High-fidelity character autonomy for virtual small unit training," in Interservice/Industry Training, Simulation, and Education Conference, Orlando, 2012.

[Sun21] Zhiqing Sun, Z., Cao, S., Yang, Y., Kitani, K.M. (2021). Rethinking Transformer-Based Set Prediction for Object Detection. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 3611-3620

[Tenzer22] Tenzer, M., Rasheed, Z., Shafique, K., & Vasconcelos, N. (29 Jun 2022). Meta-Learning over Time for Destination Prediction Tasks. arXiv:2206.14801v1 [cs.LG]. Ashburn VA.

[Towell90] Towell, Geofrey G., Jude W. Shavlik, and Michiel O. Noordewier. "Refinement of approximate domain theories by knowledge-based neural networks." In Proceedings of the eighth National conference on Artificial intelligence, vol. 861866. 1990.

[Towell93] Towell, Geoffrey G., and Jude W. Shavlik. "Extracting refined rules from knowledge-based neural networks." Machine learning 13, no. 1 (1993): 71-101.

[Valkov18] Valkov, Lazar, Dipak Chaudhari, Akash Srivastava, Charles Sutton, and Swarat Chaudhuri. "Houdini: Lifelong learning as program synthesis." arXiv preprint arXiv:1804.00218 (2018).

[Vaswani17] A. Vaswani, S. Noam, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need," Advances in Neural Information Systems, vol. 30, 2017.

[Verma18] Verma, Abhinav, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. "Programmatically interpretable reinforcement learning." In International Conference on Machine Learning, pp. 5045-5054. PMLR, 2018.

[Verma19] Verma, Abhinav, Hoang M. Le, Yisong Yue, and Swarat Chaudhuri. "Imitation-projected programmatic reinforcement learning." arXiv preprint arXiv:1907.05431 (2019).

[Vilamala20] Vilamala, Marc Roig, Harry Taylor, Tianwei Xing, Luis Garcia, Mani Srivastava, Lance M. Kaplan, Alun Preece, Angelika Kimming, and Federico Cerutti. "A Hybrid Neuro-Symbolic Approach for Complex Event Processing (Extended Abstract)." In Proceedings of ICLP2020.

[Vilamala 23] Vilamala, Marc Roig, Tianwei Xing, Harrison Taylor, Luis Garcia, Mani Srivastava, Lance Kaplan, Alun Preece, Angelika Kimmig, and Federico Cerutti. "DeepProbCEP: A neuro-symbolic approach for complex event processing in adversarial settings." *Expert Systems with Applications* 215 (2023): 119376.

[Vilamala21] Vilamala M.R., Xing T., Taylor H., Garcia L., Srivastava M., Kaplan L. Preece A., Kimmig A., Cerutti F. (2021) Using DeepProbLog to perform Complex Event Processing on an Audio Stream. In Proceedings of the Tenth International Workshop on Statistical Relational AI (https://arxiv.org/abs/2110.08090).

[Wang22] C.-Y. Wang, A. Bochkovskiy and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-ofthe-art for real-time object detectors," arXiv, vol. arXiv:2207.02696 [cs.CV].

[Xing19] Xing, Tianwei, Marc Roig Vilamala, Luis Garcia, Federico Cerutti, Lance Kaplan, Alun Preece, and Mani Srivastava. "DeepCEP: Deep Complex Event Processing using Distributed Multimodal Information." In 2019 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 87-92. IEEE, 2019.

[Xing20] Xing, Tianwei, Luis Garcia, Marc Roig Vilamala, Federico Cerutti, Lance Kaplan, Alun Preece, and Mani Srivastava. "Neuroplex: Learning to Detect Complex Events in Sensor Networks through Knowledge Injection." In Proceedings of the 18th Conference on Embedded Networked Sensor Systems, pp. 489-502. 2020.

[Xing21] Xing, T., Garcia, L., Cerutti, F., Kaplan, L., Preece, A., & Srivastava, M (2021). "DeepSQA: Understanding Sensor Data via Question Answering." In Proceedings of the International Conference on Internet-of Things Design and Implementation, pp. 106-118.

[Xu18] Xu, D., Nair, S., Zhu, Y., Gao, J., Garg, A., Fei-Fei, L. and Savarese, S., 2018, May. Neural task programming: Learning to generalize across hierarchical tasks. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3795-3802). IEEE.

[Zhu21] Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2021, pp. 2778-2788

APPENDIX A: COMPLETE LIST OF ATOMIC AND COMPLEX EVENTS

CE1: Complex Event 1 “Prepare to Destroy a Bridge” is concluded to have occurred when all the following AEs are satisfied. Occurrence of all these AEs takes place over a period of 5-10 minutes and involves detection of objects monitored by 3 different sensors.

- 1.1.z. A formation of reconnaissance vehicles moves through a sensor area toward a bridge
- 1.1.a. A formation of reconnaissance vehicles approach a bridge in the same sensor area
- 1.1.b. A formation of reconnaissance vehicles take position on both sides of a bridge, obstructing the road at a distance of >100m from the ends of the bridge, so that POL traffic is blocked from entering the bridge.
- 1.1.c. The formation on the far side of the bridge from the tank column moves away from the bridge

The event as modeled for present purposes does not include the actual demolition of the bridge, only the performance of all the AEs that serve as prelude to the TTP in which the bridge would be demolished by detonation.

CE2: Complex Event 2 “Move a Column of Tanks Across a Bridge” is concluded to have occurred when all the following AEs are satisfied. Occurrence of all these AEs takes place over a period of 5-10 minutes and involves detection of objects monitored by 3 different sensors.

- 2.1.z A formation of reconnaissance vehicles moves through a sensor area toward a bridge
- 2.1.a. A formation of reconnaissance vehicles approach a bridge in the same sensor area
- 2.1.b. A formation of reconnaissance vehicles take position on both sides of a bridge, taking position off the road, close to the ends of the bridge (within 100m of its ends)
- 2.1.c. The tank column is detected moving toward the bridge.

This combination of AEs serve as an indication that the column of tanks intends to cross the bridge.

CE3: Complex Event 3 “Enemy forces attempt to set up a stationary defensive position” is concluded to have occurred when all the following 3 AEs are satisfied.

- 3.0.a. A formation of tank and reconnaissance vehicles enters a sensor camera box with reconnaissance vehicles >10 seconds ahead of the tank column.
- 3.1. The reconnaissance vehicle formation divides into groups moving away from the tank column by all available roads, taking up position between the tank column and any oncoming traffic
- 3.2 The tank column leaves the road and takes a defensive position off-road in a line or an arc with all tanks <10m from each other pointing turrets in the same direction.

This combination of AEs serve as an indication that the column of tanks is preparing a defensive stationary position.

When all AEs underlying a given CE have occurred, that CE is triggered, indicating that the NeuroPlex-enabled system should warn the hypothetical analyst in our use case of enemy intentions or actions to be relayed to higher authority.

APPENDIX B: SCENARIO VISUALIZATION EXAMPLES

Recall that the Phase I scenarios were about making recognitions about red forces moving armor/tank columns in and around contested bridges. The phase I use case simulated a defense by blue forces of a series of bridges in the Ann Arbor MI metropolitan area. The original scenario included the decisions regarding blue force deployment in response to predictions of enemy intentions when moving through three different named areas of interest (NAI). A limited subset of this scenario was implemented as the Phase I AE/CE recognition challenge, using three simulated fixed position aerial surveillance assets described below in Figure 5 (repeated).

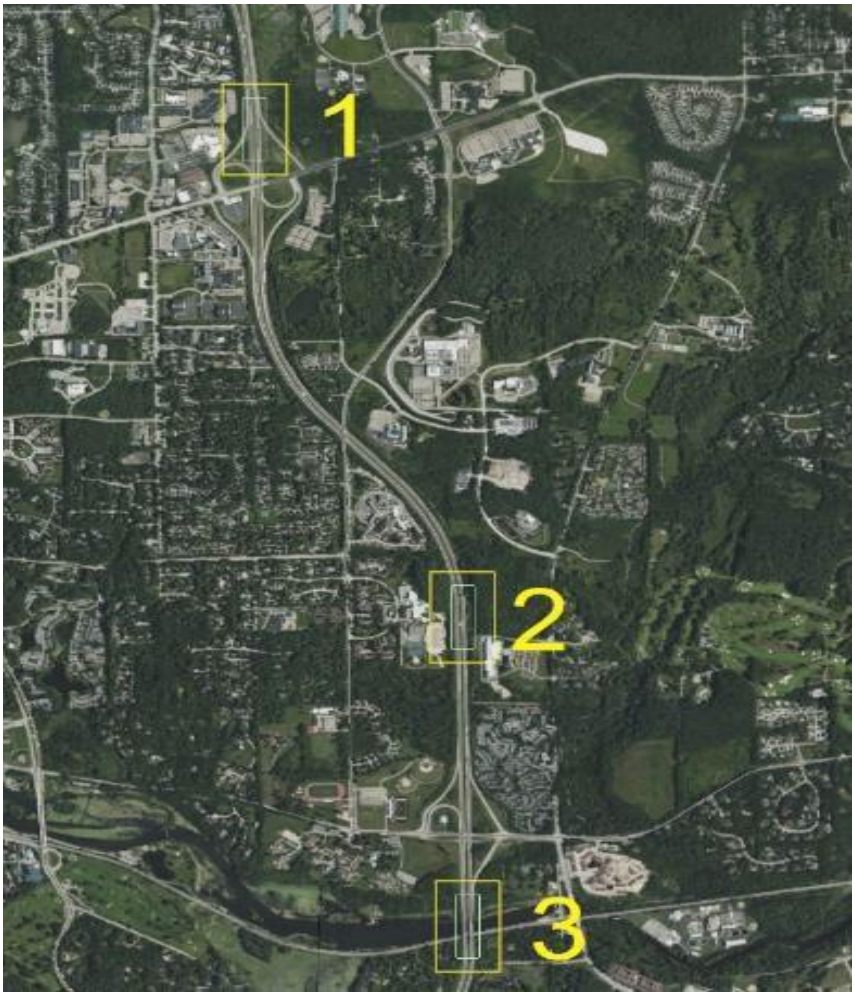

Figure 5. Fixed sensor positions

As depicted in Figure 8, Sensor 1 was modeled as two concentric sensor viewing boxes directly north of Highway 23 of dimensions 500m x 280m and 350m x 100m. Sensor 2 (see Figure 9) was modeled using the same structure at a position 2000m north of the Enborne River, and Sensor 3 (Figure 10) was modeled as a viewing area of the same size, with watchboxes defined on the north and south sides of the river used to monitor for reconnaissance vehicles in advance positions.

Figure 8. Sensor Viewing Area 1

Figure 9. Sensor Viewing Area 2

Figure 10. Sensor Viewing Area 3

We defined three different CEs involving movement of a column of tanks and recce vehicles through Ann Arbor toward a bridge (see Appendix A). In all CEs, the column consisted of 4-6 tanks moving in two staggered columns with targeted separation of 20m between the nearest neighbor when in formation. The column was escorted by a set of 4 reconnaissance vehicles that moved in pairs, with targeted separation from the tank column of two minutes, and a targeted distance between the two pairs of reconnaissance vehicles of 50m during travel. All roads were also populated by a randomly distributed set of 400-500 civilian vehicles.

Figure 11. Sensor 1 images of recce vehicles and tank convoy.

Figure 12. Sensor 3 images of recce vehicles and tank convoy crossing bridge.

Figure 13. Views of civilian vehicles from two sensors.