ARL-MR-1142 ● Nov 2025

Learning from Demonstration in a Military Drill: Artificial Intelligence of Maneuver and Mobility – Human Agent Teaming Integrated Capability Experiment (AIMM-HAT ICE)

by Angelique Scharine, David Chhan, Brian Cesar-Tondreau, Vernon Lawhern, Joe Rexwinkle, and Julia L. Wright

NOTICES

Disclaimers

ARL-MR-1142 ● Nov 2025

Learning from Demonstration in a Military Drill: Artificial Intelligence of Maneuver and Mobility – Human Agent Teaming Integrated Capability Experiment (AIMM-HAT ICE)

Angelique Scharine, David Chhan, Brian Cesar-Tondreau, Vernon Lawhern, Joe Rexwinkle, and Julia L. Wright
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
November 2025	Memorandum Report	10/1/2024	9/30/2025

4. TITLE AND SUBTITLE
Learning from Demonstration in a Military Drill: Artificial Intelligence of Maneuver and Mobility – Human Agent Teaming Integrated Capability Experiment (AIMM-HAT ICE)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Angelique Scharine, David Chhan, Brian Cesar-Tondreau, Vernon Lawhern, Joe Rexwinkle, and Julia L. Wright

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-FD Aberdeen Proving Ground, MD 21005	ARL-MR-1142

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

ORCIDs: Angelique Scharine, 0000-0001-8230-1824; David Chhan, 0000-0003-2470-8663; Brian Cesar-Tondreau, 0000-0002-6309-7001; Vernon Lawhern, 0000-0002-3921-8723; Joe Rexwinkle, 0000-0002-3856-101X; Julia L. Wright, 0000-0003-3026-1538

14. ABSTRACT

The Artificial Intelligence of Maneuver and Mobility – Human Agent Teaming Integrated Capability Experiment (AIMM-HAT ICE) explored novel autonomous capabilities through a three-part experiment. AI navigation models typically require large datasets for training and modeling expertise. However, Learning from Demonstration (LfD) techniques offer the possibility for the regular Soldier, with no prior coding or ML experience, to customize training by demonstrating the desired behavior. To show the utility of these techniques for autonomous vehicles, a simulated Warthog was trained to perform a military maneuver known as a "berm drill" at Grace's Quarters Robotics Research Collaboration Campus (R2C2). Data was collected from both a simulated (Unity game engine) environment and at R2C2 during February and March 2025. The final project was presented as part of the successful AIMM-HAT ICE demonstration on 11 September 2025.

15. SUBJECT TERMS
artificial intelligence, learning from demonstration, autonomy, unmanned ground vehicles, berm drill, Humans in Complex Systems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	28

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Angelique Scharine	(520) 691-5562

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction and Background

1.1 Armored Autonomy

Advances in artificial intelligence (AI), exemplified by self-driving cars, suggest potential for automating tasks and reducing workload and errors in various applications (Wishart et al. 2023). These technologies are applicable to both manned and unmanned armored vehicles, across ground and aerial platforms. However, training armored vehicles for complex military maneuvers presents challenges due to the lack of contextual predictability required for fully autonomous operation (Rademaker and Mevissen 2022; Kascha and Henze 2023).

1.2 Learning from Demonstration

Learning from Demonstration (LfD) is a technique in which agents learn to perform tasks by observing and mimicking a human expert. A simple analogy is updating a spell checker's dictionary to recognize new words—effectively altering the system's behavior through demonstration.

LfD encompasses various methods. Behavior cloning (or imitation learning) directly maps observed states to actions, replicating the demonstrator's behavior (Schaal 1999). Alternatively, inverse reinforcement learning infers the reward function driving the expert's actions, allowing the agent to learn the underlying goals (Ng and Russell 2000). The core objective is to enable robotic agents to learn policies through supervised learning, using demonstration data as labeled examples or to shape the reward function (Argall et al. 2009; Ho and Ermon 2016; Osa et al. 2018).

LfD has been applied to autonomous vehicles successfully to train such behaviors as maneuvering through a parking lot (Abbeel et al. 2008) and making decisions at intersections (Wu et al. 2022). It has even been used to train robot navigation in unstructured terrains (Silver et al. 2010).

This research aims to empower Soldiers to train robotic assets without specialized technical expertise. We demonstrate the feasibility of training an unmanned ground vehicle (UGV) to perform a basic battle maneuver in a simulated environment.

1.3 Application of LfD to Autonomous Navigation

LfD shows promise for adapting autonomous navigation to challenging environments. However, learned behaviors may be context-dependent and potentially detrimental in other situations. Xiao et al. (2020) addressed this by combining a segmenting algorithm with behavioral cloning, enabling the system to identify appropriate behaviors for specific contexts. They successfully trained a Clearpath Jackal on a course with varying terrains—curves, obstacles, narrow corridors, and open spaces—and achieved improved navigation on each segment.

Behavior cloning and imitation learning can also enhance pre-existing navigation models. The U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) Autonomy Stack (DEVCOM Army Research Laboratory 2025) uses an off-the-shelf navigation package with local and global planners. César-Tondreau et al. (2021) demonstrated that integrating a behavior cloning module, trained on navigation demonstrations, between the global and local planners resulted in navigation behavior more closely resembling the human demonstration.

Furthermore, LfD can improve navigation in terrains where standard sensors are inadequate. For example, conventional navigation systems relying on cameras, light detection and ranging (LiDAR), and thermal sensors struggle to assess the depth of negative obstacles (e.g., grooves, dips, and holes). César-Tondreau et al. (2022) improved obstacle negotiation by incorporating data from an inertial measurement unit (IMU) into the demonstration data.

2. Robotic Platform and Data Collection for Berm Drill Training

2.1 The Warthog UGV

The Warthog is a rugged, all-terrain UGV designed for military, industrial, and research applications (Figure 1). Its robustness and versatility allow operation in complex and harsh environments. The Warthog is equipped with advanced autonomous navigation systems, such as GPS, LiDAR, and computer vision, and it can operate independently or with minimal human intervention. For this demonstration, the UGV was outfitted with LiDAR, a front-facing camera, and a pan-tilt-zoom (PTZ) sensor on a stalk.

Figure 1. Warthog UGV.

2.2 Hardware and Software Components

To simulate the expert demonstrations that a Soldier might use to train a UGV, the Warthog was teleoperated using the Warfighter–machine interface (WMI) (Figure 2). Video data was transmitted wirelessly from the UGV's front-facing and PTZ cameras. Initially, network communication was provided by a Silvius radio system (Figure 3). A separate computer recorded all sensor data from both the UGV and the WMI during teleoperated demonstrations. Although some of the data were collected in this manner, because wireless data transmission was slow and unreliable, most of the demonstration trials were obtained by directly teleoperating the UGV using its remote control.

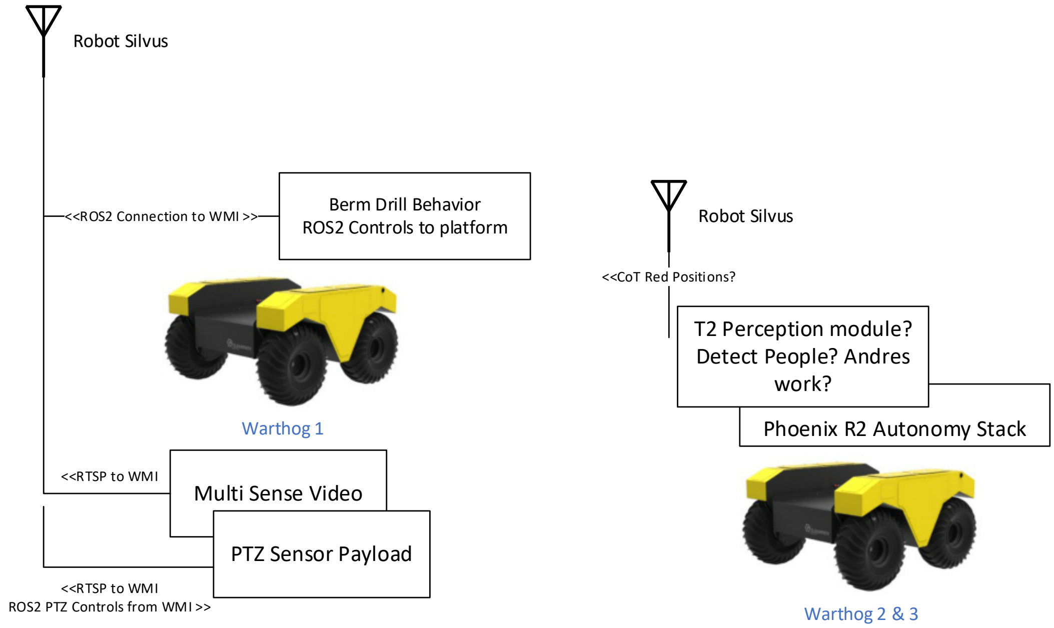

Figure 2. Hardware and software used in the experiment.

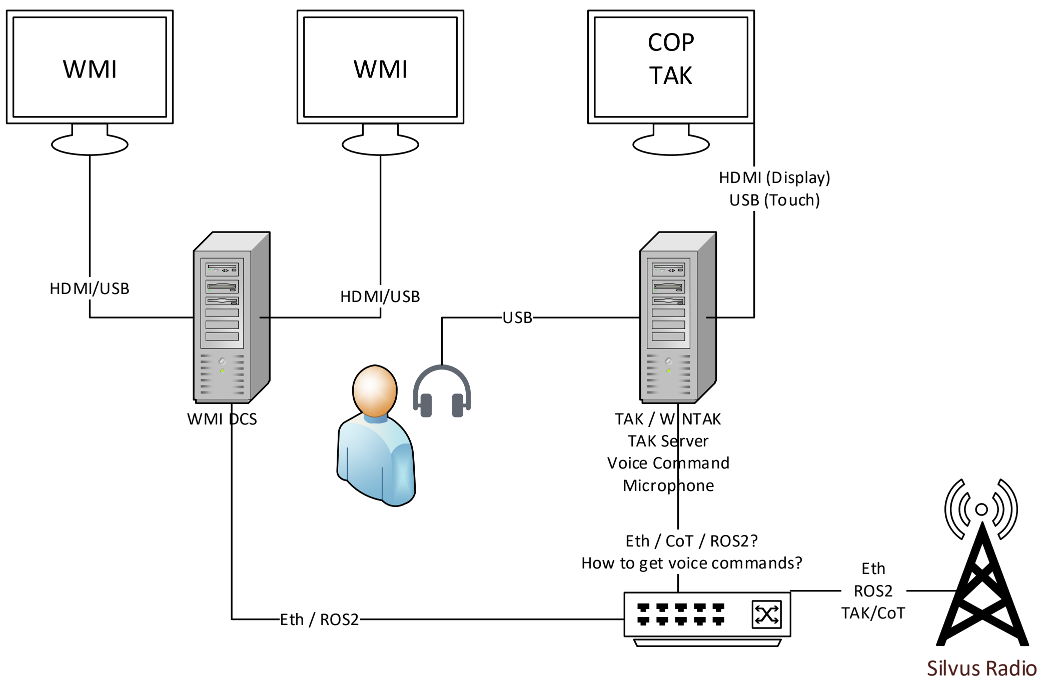

Figure 3. Deployment architecture.

2.3 Robot Task

2.3.1 Berm Drill Task Analysis

Berms provide crucial cover and concealment for soldiers (Figure 4), enabling suppressive fire while minimizing exposure (Marine Corps. 2002; NATO 2004; DOA 2011). Berm drills are collective tasks that require coordination between multiple team members. A cognitive task analysis (CTA) of the berm drill maneuver was conducted to better understand the task roles, subtasks, and key information requirements. After the CTA was completed, it was expanded to include autonomous tasking, either wholly or in part, as well as identifying those tasks/subtasks that would be inappropriate to assign to an autonomous system (e.g., subjective decision point). Sources of information to support autonomous actions were identified, and the task was further subdivided into potential autonomy actions. Full details of the CTA findings are reported in Johnson et al. (2024).

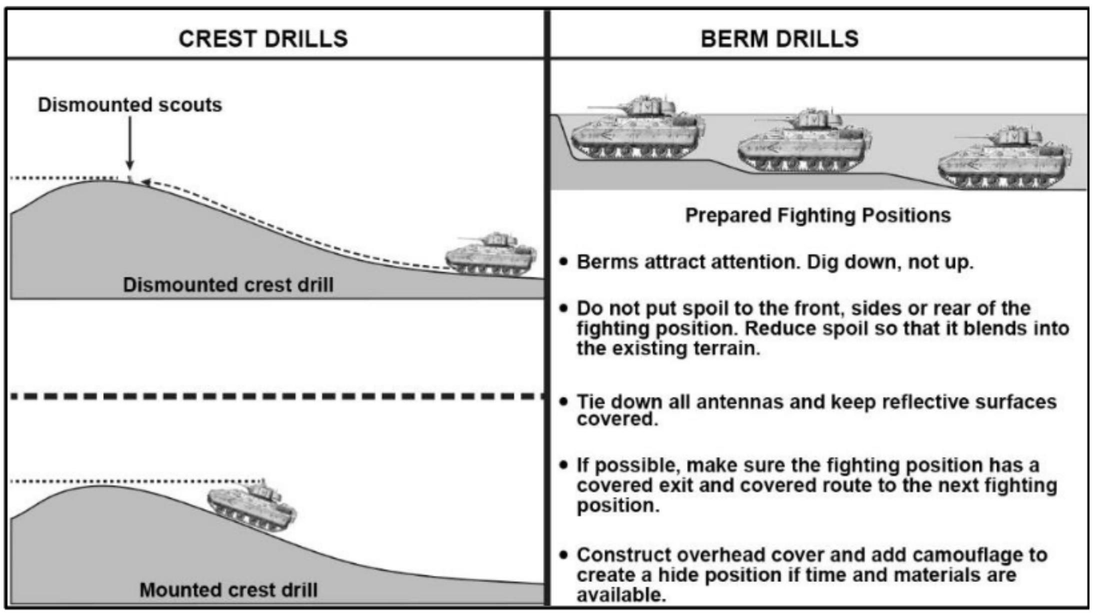

Figure 4. Diagram of a berm or crest drill (DOA 2019).

2.3.2 Berm/Crest Drill Training

The berm/crest tactic has been used since World War I and allows dynamic maneuver and sustained firepower. For training, the berm drill task was divided into three phases: approach, firing, and retreat, guided by crew speech indicating the active phase.

The UGV was trained using crew speech, LiDAR data, front-facing camera data, and data from the PTZ sensor. Assuming known opposing force (OPFOR) location (via intelligence, UGV sensors, or unmanned aerial vehicle assets), the objective was to train the UGV to navigate from a hull-down position to a firing position, acquire the target, and retreat.

2.4 Data Collection Sites

Data was collected at two locations, in a Unity based simulation and at the ARL Robotics Research Collaboration Campus (R2C2)

2.4.1 ARL Unity-Based Simulation Testbed

This simulation environment, integrated with the ARL's Phoenix Autonomy Stack, allows virtual training using robotic operating system (ROS) commands and generates sensor data (IMUs, camera, and LiDAR).

2.4.2 Robotics Research Collaboration Campus

Live data was collected at the R2C2 facility (Figure 5), but ultimately this was not incorporated into the final training model (see Section 2.5.1.2).

Figure 5. R2C2 military operations on urban terrain (MOUT) site.

2.5 Data Collection

The UGV was trained in three stages:

Stage 1. Simulated Navigation: Training the UGV to navigate to a location using sensor data within the simulated environment.
Stage 2. Simulated Berm Drill: Training the UGV to perform the forward and reverse movements required for the berm drill maneuver within the simulated environment, separated into the approach, firing, and retreat phases.
Stage 3. Live Training Trials: Collecting real-world data at R2C2 to refine and augment the simulated data.

2.5.1 Simulated Data

The simulation used a replica of the R2C2 site within the Unity game engine, leveraging the AIMM/ros2/arl-warthog-sim package from ARL's Phoenix Autonomy Stack (DEVCOM Army Research Laboratory 2025). A simulated Warthog represented the blue (friendly) force (BLUFOR), and a Russian T90 tank served as the OPFOR. Data collection included planar LiDAR scans, velocity commands, camera images, IMU data, OPFOR coordinates, and visibility angles (Table 1). Data was recorded at 2 Hz and saved to comma-separated value (CSV) files. OPFOR detection was constrained to a 145° field of view, whereas BLUFOR possessed a 360° detection capability within its designated radius.

Table 1. Topic names and types for the sensor data recorded.

Topic name	Topic type
_warty_odometry_global	nav_msgs/msg/Odometry
_warty_odometry_local	nav_msgs/msg/Odometry
_warty_platform_odom	nav_msgs/msg/Odometry
_wmi_mobility_dataLogger	std_msgs/msg/String
_wmi_weapon_dataLogger	std_msgs/msg/String
_warty_cmd_vel	geometry_msgs/msg/Twist
_tf	tf2_msgs/msg/TFMessage
_warty_sensors_microstrain_compass	sensor_msgs/msg/Imu
_warty_sensors_ptz_joy	sensor_msgs/msg/Joy
_warty_sensors_geofog_imu_data_enu	sensor_msgs/msg/Im
_warty_sensors_zoom_cam_camera_info	sensor_msgs/msg/CameraInfo
_warty_sensors_multisense_front_left_depth_camera_info	sensor_msgs/msg/CameraInfo
_warty_sensors_geofog_kvh_raw_gnss	kvh_geo_fog_3d_msgs/msg/KvhGeoFog3DRawGNSS
_warty/scan	sensor_msgs/msg/LaserScan

2.5.1.1 Stage 1: Navigation Only

Before berm drill training, the UGV was trained on basic navigation. The UGV was placed in a random location and assigned a target location within 20–50 m. It then navigated a collision-free trajectory to the target, recording sensor data.

2.5.1.2 Stage 2: Simulated Berm Drill

The berm drill task was separated into three phases:

1. Driving from a hull-down position while scanning for a target.
2. Sighting and "shooting" the target (capturing an image with the PTZ camera).
3. Returning to the hull-down position.

The UGV started within 50 m of the OPFOR, navigated until the target was visible, and then returned to its starting location.

2.5.2.3 Stage 3: Training Data Collection March–April 2025

Live data was collected at R2C2 to refine the simulated data. The same phases as the simulated berm drill were used, with the UGV teleoperated via the WMI. Sensor data related to location, velocity, WMI navigation, camera, and LiDAR input was recorded. Four targets were used, and their coordinates varied between testing dates (Table 2). Four targets were placed on the MOUT site (see Table 2 and Figure 6).

Table 2. Target locations.

	24 March 2025		16 April 2025	
	Latitude	**Longitude**	**Latitude**	**Longitude**
1	N 39 21.061	W 076 20.640	N 39 21.037	W 076 20.386
2	N 39 21.062	W 076 20.702	N 39 21.062	W 076 20.702
3	N 39 21.062	W 076 20.423	N 39 21.062	W 076 20.423
4	N 39 21.057	W 076 20.759	N 39 21.027	W 076 20.449

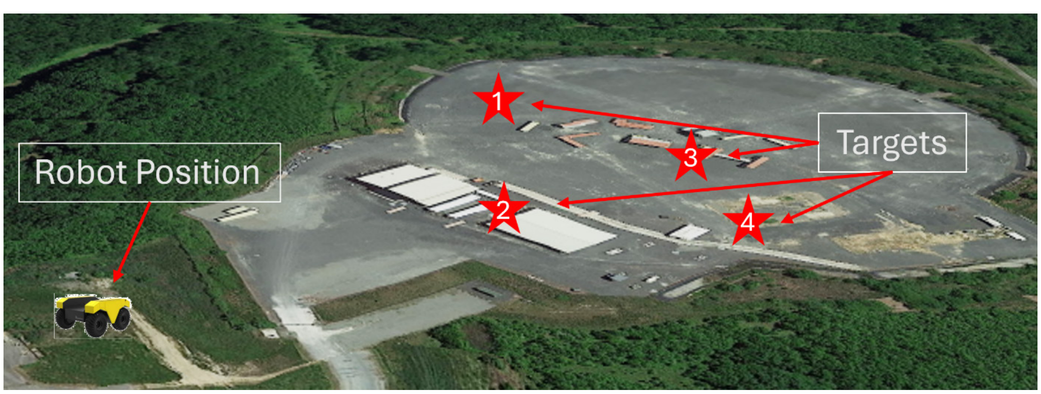

Figure 6. Depiction of the target locations and the Warthog location.

Trials began with the UGV oriented ±90° from the sight point to force target orientation. Nine starting points, approximately 5 ft apart, were identified and marked with flags (Figures 7 and 8). Before each trial, the UGV was navigated until the target was visible in the PTZ sensor's video feed (Figure 7, top). This location was marked with a cone. The UGV was then returned to its starting point, and the trial began.

During each trial, the UGV was teleoperated from a hull-down start position to a sight point where the target was visible in the PTZ camera feed. To prevent the UGV from simply learning to move forward, it was initially oriented ±90° from the sight point (Figure 7, bottom), which required it to orient itself towards the target. The sight point locations varied depending on the target and starting point locations, with distances ranging from 3 to 92 ft.

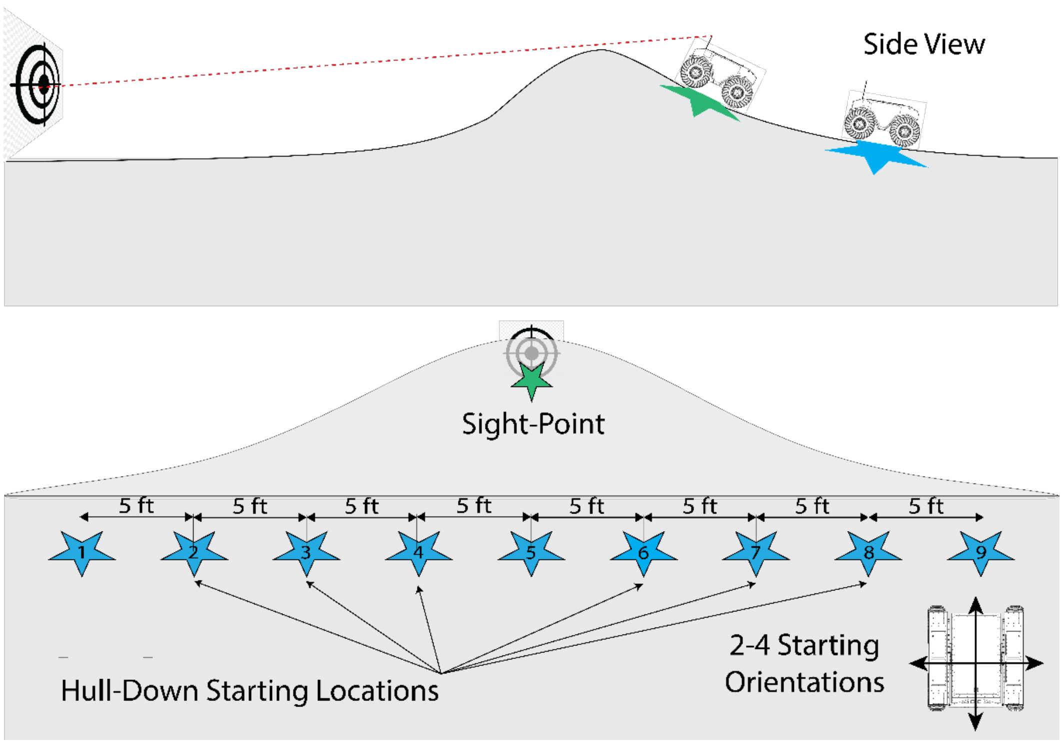

Figure 7. Schematic showing the different test conditions for data collection. (Top panel) UGV can see the target from the green sight point and can also be seen from the target. The UGV has cover from the target when positioned at the hull-down position. (Bottom panel) The berm was conducted from each of 10 initial starting points and each of four starting orientations.

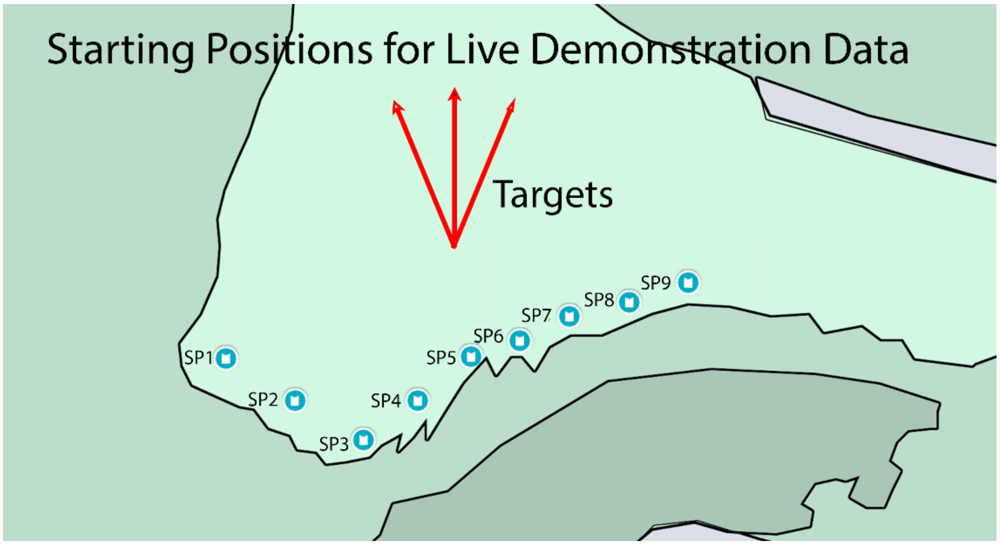

Figure 8. **Map of live test site and positions of start points.**

Initially, it was assumed that the model could identify sight points by finding the nearest unobstructed view of the target using topological data and line-of-sight computations. However, the collected data was ultimately not used for the final model training due to the following:

- Difficulty identifying the target site point in the real world due to obstructions such as grass and trees.
- The model ignored LiDAR input due to the lack of real-world obstacles in the live data.
- Recognizing that leveraging existing navigation programs for individual steps, and training the model on the sequence of steps, would be more effective. This would involve the Soldier specifying the site point and instructing the robot to navigate, fire, and return.

2.6 Training

2.6.1 Pretrained Basic Navigation Policy

Our approach in training up a basic navigation policy modeled prior work in using imitation learning and behavioral cloning techniques to improve autonomous robot navigation (César-Tondreau et al. 2021). Our pretrained navigation pipeline takes three separate inputs: the planar LiDAR scan, remaining distance and heading to the goal location, and an image from the forward-facing camera; trained with a deep neural network classifier (Figure 9). The action space consists of nine equally spacing discretized linear velocity and angular velocity.

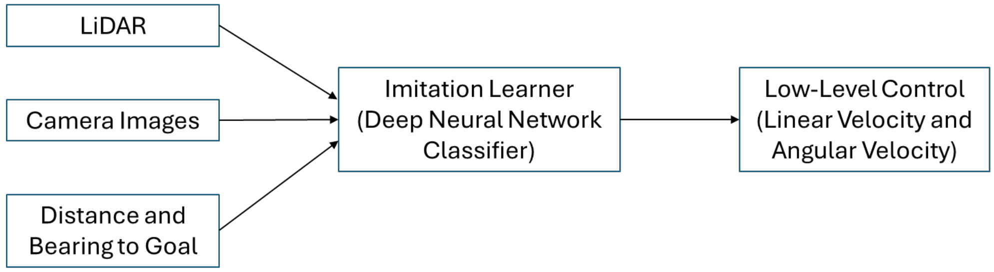

Figure 9. Navigation training pipeline.

2.6.2 Model Inputs and Outputs

The LiDAR input includes a 360° planar LiDAR scan at the robot's center. The scan comprises 722 IR beams, each of which is spaced 5° from each other and has a maximum range of 30 meters from the robot's center. This data is stored in a 1×722 element vector. The remaining distance and offset heading to the goal location are represented as a 1×2 vector. The image input is a $640 \times 480 \times 3$ color image from the forward-facing camera.

We used a temporal stacking technique to the above-mentioned features that stack the features observed in the current time step with those from the previous timestep. This technique avoids state aliasing by providing the model with historical context for the demonstrated actions.

The local navigation output is represented by the robot's linear and angular velocity. The range of these action spaces was categorized into nine discrete values that served as the output for our neural network models.

2.6.3 Network Architecture

We trained a deep network classifier* that takes in three separate inputs: the planar LiDAR scan, remaining distance and heading to the goal location, and an image from the forward-facing camera, all of which are detailed in the previous section (Table 3). The LiDAR input is passed through two sequential fully connected layers, which have 128 and 64 hidden units, respectively. Each layer has a rectified linear unit (ReLU) activation and is followed by a 50% dropout regularization layer. The image input is passed through three consecutive 2D convolutional layers, which have 16, 32, and 64 hidden units, respectively. Each of the convolutional layers has a 5×5 kernel size and ReLU activation. In addition, they are followed by a 2D Average Pooling layer with a 3×3 kernel size and a Dropout layer with 50% regularization. The resulting feature vector from the third convolutional layer is flattened and concatenated with the output vector from second fully connected

* The code for this research is hosted in a GitLab Repository. Should you wish to obtain this code, please contact the authors.

layer and the raw Goal and Heading input vector. The concatenated vector is passed through three fully connected layers with 576, 192, and 64 hidden units, respectively, before finally being passed through the fully connected output layer with nine hidden units and SoftMax activation.

Table 3. Summary of training model.

Input block (layer)	Layer	Filter	Size	Parameters	Output	Activation
Image (1)	Conv2dD AvgPool2D Dropout (5%)	2 × 16	5 × 5	816	16 × 74 × 74 16 × 24 × 24	ReLU
Image (2)	Conv2dD AvgPool2D Dropout (5%)	16 × 32	5 × 5 3	12,832	32 × 22 × 22 32 × 7 × 7	ReLU
Image (3)	Conv2dD AvgPool2D Dropout (5%)	32 × 64	5 × 5 3	51,264	32 × 5 × 5 32 × 1 × 1	ReLU
LiDAR (1)	Linear Dropout (5%)	722 × 128		92,544	2 × 1 × 128	ReLU
LiDAR (2)	Linear Dropout (5%)	128 × 64		8,256	2 × 1 × 64	ReLU
LiDAR-image-goal concatenated (1)	Linear	196 × 576		113,472	1 × 576	ReLU
LiDAR-image-goal concatenated (2)	Linear	576 × 192		110,784	1 × 192	ReLU
LiDAR-image-goal concatenated (3)	Linear	192 × 64		12,352	1 × 64	ReLU
Output	Linear	64 × 9		585	1 × 9	SoftMax

2.7 Training Results

We trained separately two neural network models to predict linear velocity and angular velocity given the input sensor data (LiDAR, remaining distance and heading to goal, and camera image). Figures 10 and 11 showed confusion matrices of the linear velocity and angular velocity model predictions. Although the models performed well with the weighted F1[*] scores of 0.85 and 0.87 in correctly classifying the actions, when testing in the simulation, the models failed to consistently navigate around the obstacles. The robot was able to navigate from the starting point to goal location but was running into obstacles during some trials.

[*] F1 = 2(Precision*Recall)/(Precision + Recall)

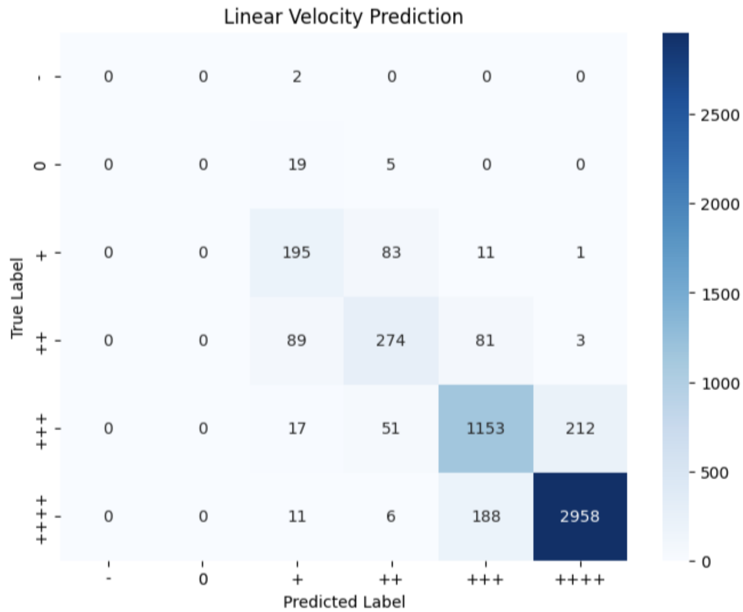

Figure 10. Confusion matrix of the predicted vs. true linear velocity classification with the weighted F1 score of 0.85.

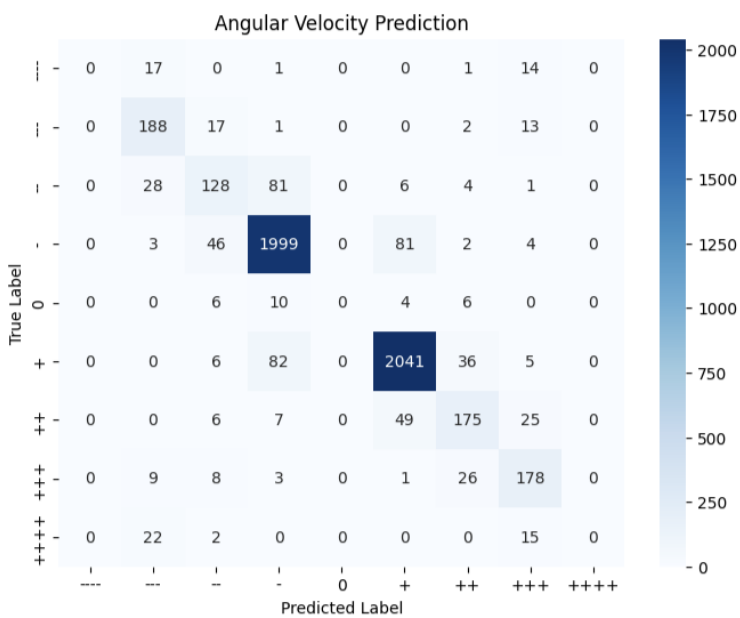

Figure 11. Confusion matrix of the predicted vs. true angular velocity classification with the weighted F1 score of 0.87.

Our initial approach was to pretrain a base navigation policy using behavioral cloning technique with the navigation-only simulated data, then use the basic navigation policy to learn berm drill behaviors from the simulated berm drill data, and finally to perform transfer learning of the berm drill behaviors from the simulation to the real-world data. With the inconsistency in our pretrained navigation policy and time constraint, we decided to change course in switching the learned local navigation to using the built-in navigation algorithms of the ARL Autonomy Stack instead. In the next section, we describe how the berm drill was automated using this new approach.

2.8 Executing Automated Berm Drill Behavior

Executing a berm drill requires coordinated action between the gunner and the driver. We aimed to automate the driving behavior of the berm drill to free up the driver to perform other critical tasks. Just like actions taken by the driver in a berm drill, our automated berm drill algorithm can be implemented as sequences of driving behaviors triggered by an event (i.e., gunner's command) (Table 4).

Table 4. Driving behaviors and triggers.

Driving behavior	Trigger
Drive up	Voice activated or button press
Hold for fire	Robot and target establish line of sight
Drive down	Voice activated or button press

We assume that, in practice, we are given the coordinates of the OPFOR target from a scouting aerial drone and the "visibility_metrics" code provides locations on the map with visual line of sight to the target. In simulation, we are given the latter location for the robot to navigate.

When executing the berm drill behavior, the program listens to key words in the command phrase spoken by the human operator. Each command phrase activates one of four movement behaviors corresponding to the phases in the berm drill maneuver:

- Phrases containing "orient," "turn," or "face" such as "turn to target" will have the robot orient in place such that its forward heading is in the direction of the target's location.
- Phrases containing "up," "forward," or "face," such as "Drive up to the target," will have the robot drive up the berm, while maintaining its heading towards the target, until it crests the berm out of cover.
- Phrases containing "wait," "pause," or "halt" such as "halt movement" will interrupt whatever phase of the berm drill the robot is currently executing.
- Phrases containing "return," "back," or "reverse," such as "return to origin," will have the robot drive backward down the berm while maintaining its back heading toward the starting point.

The robot starts at the base of the berm, and the user will give the "turn to target" command followed by the "forward" command. The robot will travel up the berm, maintaining its heading in the target's direction, until the user has a visual on the target where they will give the "halt" command. The user will then give the "reverse" command to move back behind cover or return to its starting location at the base of the berm.

This procedure was implemented in the Unity simulation engine for demonstration purposes. Time constraints prevented its implementation with the Warthog, but it is presumed that it would function similarly. One caveat is that the viewshed calculator does not account well for the effects of trees and vegetation. The operator may need to adjust or preselect the firing position based on real-world feedback from the PTZ camera.

3. Results and Discussion

Using a combination of built-in navigation models and breaking the task down into its component parts, we were able to train the model to complete the berm drill task autonomously. However, we did not use LfD, because the model was not trained on demonstrations of appropriate behavior. Instead, we used a combination of previously trained models and cognitive task analysis. Through task analysis, waypoints were identified, and the automated behavior was achieved by stringing together preprogrammed navigational behavior with the identified waypoints.

Because our objective was to demonstrate the usefulness of LfD, it is worth considering whether LfD is a viable option for other forms of Soldier-guided adaptation (SGA). LfD has worked in other instances (Silver et al. 2010, 2013).

The base policy succeeded in navigating to any goal location but failed to avoid obstacles along its path. We suspect that the model was failing to learn from the LiDAR input data, because most of the paths between starting and ending positions have the robot traversing empty space where the obstacles are distant enough for the robot to still have a straight-line path to the goal. One approach to address this issue would be to create a simulated environment where the obstacle density makes straight-line trajectories between the start and goal positions significantly less often. Thus, this forces the model to use the LiDAR data when navigating. If we had time to retrain the model with sufficient instances requiring avoidance of obstacles, it is highly likely that this behavior would have been learned (as in César-Tondreau 2022).

Another factor is the issue of determining an appropriate firing location. One of the primary objectives of the berm drill procedure is minimizing the time during which the OPFOR can see the BLUFOR vehicle. Therefore, it is unlikely to be helpful to manually identify the location. In addition, GPS topology information is unlikely to contain information about vegetation, making its usefulness for viewshed calculation limited. One possibility is to use computer vision to identify when the target OPFOR is within range. This is not currently sophisticated enough to automate but is likely to improve within the time it would take to develop LfD SGA systems.

Another possibility is that SGA be used in exactly the way it was achieved here. The WMI already incorporates automated navigation using waypoints and aided/automatic target recognition (DEVCOM Ground Vehicle Systems Center 2025). Perhaps LfD could consist primarily of indicating waypoints, some of them fixed and others dependent on visibility criteria. Interestingly, this is the approach taken by Silver et al. (2010). Their approach was different from standard imitation learning in that the expert was providing the data labeling and task analysis for the demonstration.

4. Conclusions

In conclusion, this work highlights both the potential and limitations of LfD techniques for customizing autonomous vehicle behavior. Although training a model through demonstrated examples is feasible, its effectiveness is constrained by the scope of the training data and the quality of the vehicle's sensor input. Currently, leveraging established navigational models for primary movement remains preferable. Implementing LfD is best suited for defining tasks as a series of discrete, well-defined waypoints.

Furthermore, although topological data can indicate potential line of sight, it cannot account for visual obstructions like foliage or trees. A computer vision system could aid in target acquisition, but it introduces the risk of exposing the vehicle. Therefore, positive target identification and confirmation of clear visibility must still be performed by a human observer before engagement. Despite these limitations, automating these procedures can reduce the cognitive load on the operator, enabling them to focus on maintaining situational awareness and security while the UGV maneuvers.

5. References

Abbeel P, Dolgov D, Ng AY, Thrun S. Apprenticeship learning for motion planning with application to parking lot navigation. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2008; Nice, France. p. 1083–1090. https://doi.org/10.1109/IROS.2008.4651222

Argall BD, Chernova S, Veloso M, Browning B. A survey of robot learning from demonstration. Robotics and Auton Systms. 2009;57(5):469–483.

César-Tondreau B, Warnell G, Kochersberger K, Waytowich N. Towards fully autonomous negative obstacle traversal via imitation learning based control. Robotics. 2022;11:67. https://doi.org/10.3390/robotics11040067

César-Tondreau B, Warnell G, Stump E, Kochersberger K, Waytowich N. Improving autonomous robotic navigation using imitation learning. Front Robot AI. 2021;8:627730. https://10.3389/frobt.2021.627730

DEVCOM Army Research Laboratory. Phoenix-R2 autonomy software stack [GitLab internal repository]. DEVCOM Army Research Laboratory (US); [accessed 2025]. https://gitlab.sitcore.net/aimm/ros2/phoenix-r2

DEVCOM Ground Vehicle Systems Center. Request for information for products and services that would enable the development of advance crewstations for the US Army's next generation of combat vehicles. DEVCOM Ground Vehicle Systems Center (US); 2025 May 8 [accessed 2025 Sep 18]. https://govtribe.com/file/government-file/coat-rfi-for-next-gen-crew-compt-tech-opsec9722-2025-05-07-dot-pdf

[DOA] Headquarters, Department of the Army. The Infantry rifle platoon and squad. Headquarters, Department of the Army (US); 2011. Field Manual No.: FM 3-21.8.

[DOA] Headquarters, Department of the Army. Scout platoon. Headquarters, Department of the Army (US); 2019 Dec. Army Techniques Publication No.: ATP 3-20.98.

Ho J, Ermon S. Generative adversarial imitation learning. Adv Neural Inf Process Syst. 2016;4565–4573.

Johnson CJ, Rexwinkle JT, Wright JL. Supporting the design of autonomous platform behaviors with an adapted cognitive task analysis. DEVCOM Army Research Laboratory (US); 2024. Report No.: ARL-TN-1229.

Kascha M, Henze R. Modular decision-making framework for level 4 applications in automated driving. 2023 29th International Conference on Mechatronics and Machine Vision in Practice (M2VIP); 2023 Nov 21–24; Queenstown, New Zealand. p. 1–6. https://doi.org/10.1109/M2VIP58386.2023.10413428

Marine Corps. Marine rifle squad. Marine Corps (US); 2002. Marine Corps Warfighting Publication (MCWP) 3-11.2.

[NATO] North Atlantic Treaty Organization. Doctrine for the Conduct of Land Operations. NATO Science and Technology Organization; 2004. NATO Standardization Agreement (STANAG) No.: 2132.

Ng AY, Russell S. Algorithms for inverse reinforcement learning. Proceedings of the Seventeenth International Conference on Machine Learning (ICML); 2000. p. 663–670.

Osa T, Piot B, Alami R, Chernova S. Algorithmic perspectives on imitation learning. Found Trends Robot. 2018;6(3–4):187–311.

Rademaker M, Mevissen S. Robotic and autonomous systems: from design to development and use in military operations. Hague Center for Strategic Studies; 2022. https://hcss.nl/wp-content/uploads/2022/11/Robotic-and-Autonomous-Systems-From-design-to-development-and-use-in-military-operations-Final.pdf

Schaal S. Is imitation learning the route to humanoid robots? Trends Cogn Sci. 1999;3(6):233–242.

Silver D, Bagnell JA, Stentz A. Learning from demonstration for autonomous navigation in complex unstructured terrain. Int J Robot Res. 2010;29(12):1565–1592. https://10.1177/0278364910369715

Silver D, Bagnell JA, Stentz A. Learning autonomous driving styles and maneuvers from expert demonstration. In: Desai J, Dudek G, Khatib O, Kumar V, editors. Experimental robotics. Springer, Heidelberg; 2013. (Springer tracts in advanced robotics, vol 88). https://doi.org/10.1007/978-3-319-00065-7_26

Wishart D, Weaver S. Apuli A. Autonomous vehicles: what are your intentions? Transp Res F. 2023;99:450–459.

Wu Z, Qu F, Yang L, Gong J. Human-like decision making for autonomous vehicles at the intersection using inverse reinforcement learning. Sensors (Basel). 2022 Jun 14;22(12):4500. https://doi.org/10.3390/s22124500

Xiao X, Liu B, Warnell G, Fink J, Stone P. APPLD: adaptive planner parameter learning from demonstration. arXiv. 2020 July 15. arXiv:2004.00116v4 [cs.RO]. https://arxiv.org/pdf/2004.00116

List of Symbols, Abbreviations, and Acronyms

2D	two-dimensional
AI	artificial intelligence
AIMM	Artificial Intelligence of Maneuver and Mobility
ARL	Army Research Laboratory
BLUFOR	blue (friendly) forces
CSV	comma-separated values
CTA	cognitive task analysis
DEVCOM	U.S. Army Combat Capabilities Development Command
GPS	global positioning system
HAT	Human Agent Teaming
ICE	Integrated Capability Experiment
IMU	inertial measurement unit
IR	infrared
LfD	Learning from Demonstration
LiDAR	light detection and ranging
MCWP	Marine Corps Warfighting Publication
MOUT	military operations on urban terrain
OPFOR	opposing force
PTZ	pan-tilt-zoom
R2C2	Robotics Research Collaboration Campus
ReLU	rectified linear unit
ROS	robotic operating system (also ROS2)
SGA	Soldier-guided adaptation
STANAG	standardization agreement
UGV	unmanned ground vehicle
WMI	Warfighter–machine interface

1 (PDF) DEFENSE TECHNICAL INFORMATION CTR
DTIC OCA

1 (PDF) DEVCOM ARL
FCDD RLB CI
TECH LIB