

467	Contents	
468	1 Introduction	1
469	1.1 Related Work	2
470	2 Background	3
471	3 Problem Statement	4
472	4 Approach	5
473	4.1 Differentiable Signal Temporal Logic for Planning	5
474	4.2 GNNs for Planning in Multi-Agent Systems	5
475	4.3 Collision Avoidance in MA Systems	6
476	4.4 End-to-End Differentiable Learning for MA-STL	6
477	5 Experiment Setup	6
478	6 Results	7
479	7 Limitations	8
480	8 Conclusion	8
481	A Implementation Details	13
482	A.1 Environment Details	13
483	B STL Specifications	14
484	C Additional Experiments	14
485	C.1 SingleIntegrator Environment	14
486	C.2 DubinsCar Environment	14
487	C.3 DoubleIntegrator Environment	14
488	C.4 Real-world Drone Experiments	15
489	D Hardness of MA-STL specifications	16

A Implementation Details

Node features and edge features Following Zhang et al. [21], the node features $v_i \in \mathbb{R}^{\rho_v}$ encode information specific to each node in our graph observation. Here, we set $\rho_v = 3$ and use the node features v_i to one-hot encode the type of the node as either an agent node, goal node or LiDAR ray hitting point node. The edge features $e_{ij} \in \mathbb{R}^{\rho_e}$, where $\rho_e > 0$ is the edge dimension, are defined as the information shared from node j to the agent at node i , which depends on the states of the nodes i and j . Since the safety objective depends on the relative positions, one component of the edge features is $p_{ij} = p_j - p_i$. The remaining edge features can be set depending on system dynamics, such as, relative velocities for double integrator dynamics.

Computation Resources All training procedures were ran on an AWS g4dn.xlarge instance or equivalent with 4 Intel Xeon-based CPU Cores and 16 GB of RAM with an Nvidia T4 GPU.

Evaluation Details Since we consider objectives that require agents to navigate close to one another at/near termination subsequently blocking the goal locations (A,B,C,D in Table 4), safety rates were reported until the point an agent had completed their plan. This can be thought of as an alternative to the agents navigating to a ‘safe’ position upon completing their specification/plan. In a drone setting, we captured this behavior by landing the drones at an agent-specific location upon completing their specification.

Planner Details For all plans, at any time step t , planning step t' , each agent i proceeded to the next waypoint $g_i(t' + 1)$ only when they reached goal $g_i(t')$ within some threshold distance $r_{goal} = 0.3$ at a time $t \geq k(t' + 1)$ where k is the goal sampling interval (Sec. 4.1). This allowed all agents to reach the waypoints in the plan without a strict time restriction on the plan duration. The asynchronous nature of our plans (among agents) fits our problem description (Defn. 1), specifically the STL Satisfaction criteria. We leave the setting where agents follow a synchronized plan to future work. For a given plan of length T with goal sample interval k (values in Table 4), the maximum trajectory length horizon during evaluation T_h was $5kT$.

A.1 Environment Details

Here, we provide the details of each experiment environment as taken from Zhang et al. [21]. We used a common simulation time step $\delta t = 0.03$ across all three environments.

SingleIntegrator We use single integrator dynamics as the base environment to verify the correctness of the implementation and to show the performance of the methods when there are no control input limits. The dynamics is given as $\dot{x}_i = v_i$, where $x_i = [p_i^x, p_i^y]^\top \in \mathbb{R}^2$ is the position of the i -th agent and $v_i = [v_i^x, v_i^y]^\top$ its velocity. In this environment, we use $e_{ij} = x_j - x_i$ as the edge information.

DoubleIntegrator We use double integrator dynamics for this environment. The state of agent i is given by $x_i = [p_i^x, p_i^y, v_i^x, v_i^y]^\top$, where $[p_i^x, p_i^y]^\top$ is the position of the agent, and $[v_i^x, v_i^y]^\top$ is the velocity. The action of agent i is given by $u_i = [a_i^x, a_i^y]^\top$, i.e., the acceleration. The dynamics function is given by:

$$\dot{x}_i = [v_i^x, v_i^y, a_i^x, a_i^y]^\top \quad (6)$$

In this environment, we use $e_{ij} = x_j - x_i$ as the edge information.

DubinsCar We use the standard Dubin’s car model in this environment. The state of agent i is given by $x_i = [p_i^x, p_i^y, \theta_i, v_i]^\top$, where $[p_i^x, p_i^y]^\top$ is the position of the agent, θ_i is the heading, and v_i is the speed. The action of agent i is given by $u_i = [\omega_i, a_i]^\top$ containing angular velocity and acceleration magnitude. The dynamics function is given by:

$$\dot{x}_i = [v_i \cos(\theta_i), v_i \sin(\theta_i), \omega_i, a_i]^\top \quad (7)$$

We use $e_{ij} = e_j(x_j) - e_i(x_i)$ as the edge information, where $e_i(x_i) = [p_i^x, p_i^y, v_i \cos(\theta_i), v_i \sin(\theta_i)]^\top$.

B STL Specifications

We formally define the Signal Temporal Logic (STL) specifications used in the experiments in Table 4. The specifications include a sequential waypoint task (*seq*), a coverage task (*cover*), a loop task (*loop*), and a branching task (*branch*). The specifications are defined over a time horizon T and are satisfied if the agents satisfy the corresponding STL formula. We use four markers A , B , C , and D to represent rectangular predicates centered around x-y coordinates $[0, 0]$, $[2, 2]$, $[2, 0]$, and $[0, 2]$, respectively. The predicates are defined as $p_i = \text{dist}(s_i, p_i) \leq 1.0$ where $\text{dist}(s_i, p_i)$ is the L1-norm ($|\cdot|_1$) distance between the agent i 's state s_i and the predicate p_i .

Spec.	Description	Formula	T	k
<i>seq</i>	Sequential of goals	$\Diamond_{[0, T/3]}(A) \wedge \Diamond_{[T/3, 2T/3]}(B) \wedge \Diamond_{[2T/3, T]}(C)$	30	20
<i>cover</i>	Coverage over goals	$\Diamond_{[0, T]}(A) \wedge \Diamond_{[0, T]}(B) \wedge \Diamond_{[0, T]}(C)$	15	20
<i>loop</i>	Loop over goals	$\Box_{[0, T/2]}(\Diamond_{[0, T/2]}(A) \wedge \Diamond_{[0, T/2]}(B))$	30	20
<i>branch</i>	Branching	$(\Diamond_{[0, T]}(A) \wedge \Diamond_{[0, T]}(B)) \vee (\Diamond_{[0, T]}(C) \wedge \Diamond_{[0, T]}(D))$	20	10

Table 4: STL specifications used in the experiments. T and k are the specification lengths and goal sample intervals respectively.

C Additional Experiments

In Tables 5, 6 and 7 we show results for the various environments and obstacle scenarios. While our GNN-ODE has an initial GNN module which can observe these obstacles, and is also trained to generate initial goals that are ‘achievable’, the GNN-ODE is inherently limited to only consider obstacles within the sensing radius R of the agents at planning time (i.e. $t = 0$). As in Zhang et al. [21], the GCBF+ controller is trained to avoid obstacles. Thus, with a robust plan, we can achieve reasonably high success rates in this setting as well due to the run-time collision avoidance maneuvers. Planning times are nearly similar to the results in Sec. 6 (Table 2) likely because our learning-based planners do not use environment dynamics at inference time and should have a similar computation cost after training is complete. For this reason, to avoid clutter, we omit this column in the following tables. We include results from the ODE ablation of our method as shown in Table 3 under the column ‘ODE’. Additional simulation videos are hosted online¹.

C.1 SingleIntegrator Environment

In Table 5 we contain the results for various combinations of specifications, 8 sampled obstacle positions (marked ‘Y’ if present, ‘N’ otherwise), and number of agents in the SingleIntegrator Environment. We observe that the GNN-ODE planner outperforms the other planners in terms of planning time and success rate across all the specifications and obstacles. We note the average improvement in success rate of 10% for our GNN-ODE planner over the MILP planner which is not as large as the improvement in the non-linear DubinsCar environment (Table 2, 6). This is due to the SingleIntegrator environment being less constrained and the MILP planner being able to find a feasible solution more easily.

C.2 DubinsCar Environment

In Table 6 we contain the results for various combinations of specifications, 8 sampled obstacle positions (marked ‘Y’ if present, ‘N’ otherwise), and number of agents in the DubinsCar Environment. On average, with obstacles present as well we get a 69% improvement in success rate for our GNN-ODE planner over the MILP planner, primarily due to the non-linear dynamics of the DubinsCar environment being challenging for the collision avoidance controller.

C.3 DoubleIntegrator Environment

In Table 7 we contain the results for various combinations of specifications, 8 sampled obstacle (Obs) positions (marked ‘Y’ if present, ‘N’ otherwise), and number of agents (N) in the DoubleIntegrator

¹Site: <https://anon-ml-git.github.io/ma-stl.github.io/>

Metric Planner			Finish Rate \uparrow			Safety Rate \uparrow			Success Rate \uparrow			TiR \downarrow		
Spec	Obs	N	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY
Branch	N	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	1000.75	396.50	257.25
		16	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	1214.50	443.00	280.87
		32	98.00	100.00	99.00	100.00	100.00	98.75	97.50	100.00	97.50	664.71	1005.81	320.23
	Y	8	100.00	93.00	98.00	100.00	100.00	95.00	100.00	92.50	92.50	1015.75	408.95	292.57
		16	99.00	96.00	94.00	97.50	100.00	95.00	96.25	96.25	88.75	1168.80	471.30	314.99
		32	97.00	98.00	94.00	98.12	100.00	92.50	95.00	97.50	86.88	1812.59	1018.77	356.09
Cover	N	8	98.00	100.00	95.00	100.00	100.00	100.00	97.50	100.00	95.00	884.86	653.00	342.93
		16	99.00	100.00	90.00	100.00	100.00	100.00	98.75	100.00	90.00	1024.40	758.25	364.56
		32	98.00	98.00	96.00	100.00	100.00	97.50	98.12	97.50	93.12	1409.89	1237.41	447.66
	Y	8	95.00	95.00	88.00	100.00	100.00	97.50	95.00	95.00	87.50	1068.46	744.83	356.17
		16	96.00	98.00	91.00	100.00	100.00	97.50	96.25	97.50	91.25	1040.65	1043.66	386.12
		32	96.00	98.00	91.00	100.00	99.38	96.88	96.25	97.50	88.75	1491.06	1297.69	488.79
Loop	N	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	1890.25	2506.00	751.00
		16	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	1445.75	2120.12	855.38
		32	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	2332.19	2635.75	1062.25
	Y	8	100.00	100.00	90.00	95.00	100.00	92.50	95.00	100.00	82.50	1434.00	2112.25	841.32
		16	99.00	96.00	93.00	96.25	100.00	96.25	95.00	96.25	88.75	2091.53	2315.44	927.83
		32	99.00	99.00	96.00	96.25	99.38	96.25	95.00	98.75	91.88	2453.19	2769.78	1137.71
Sequence	N	8	100.00	53.00	90.00	100.00	100.00	100.00	100.00	52.50	90.00	905.50	1072.30	434.44
		16	99.00	41.00	73.00	100.00	100.00	100.00	98.75	41.25	72.50	761.08	1119.07	470.17
		32	98.00	33.00	66.00	100.00	100.00	100.00	98.12	33.12	65.62	897.41	1464.39	661.21
	Y	8	98.00	45.00	88.00	100.00	100.00	100.00	97.50	45.00	87.50	731.43	1239.50	470.39
		16	98.00	36.00	62.00	100.00	100.00	100.00	97.50	36.25	62.50	1030.83	1272.50	548.39
		32	98.00	28.00	74.00	100.00	100.00	99.38	98.12	28.12	73.12	1186.13	1649.19	797.86

Table 5: Performance of different planner modules with the scalability in the number of agents (N) and specification complexity for the SingleIntegrator Environment.

Metric Planner			Finish Rate \uparrow			Safety Rate \uparrow			Success Rate \uparrow			TiR \downarrow		
Spec	Obs	N	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY
Branch	N	8	100.00	98.00	100.00	100.00	100.00	85.00	100.00	97.50	85.00	1768.75	525.57	357.00
		16	100.00	96.00	99.00	100.00	97.50	53.75	100.00	93.75	52.50	1856.12	565.09	429.81
		32	95.00	94.00	96.00	92.50	86.25	20.00	88.12	82.50	18.12	820.90	674.52	572.39
	Y	8	100.00	100.00	95.00	97.50	97.50	67.50	97.50	97.50	62.50	728.50	562.79	373.89
		16	99.00	95.00	95.00	92.50	90.00	47.50	91.25	86.25	45.00	1828.95	595.29	474.16
		32	95.00	85.00	90.00	86.88	74.38	32.50	81.88	63.75	25.00	841.66	708.91	586.89
Cover	N	8	100.00	100.00	95.00	100.00	100.00	97.50	100.00	100.00	95.00	1062.00	754.57	429.76
		16	100.00	100.00	78.00	96.25	97.50	87.50	96.25	97.50	76.25	1127.00	802.70	536.33
		32	99.00	99.00	80.00	85.00	92.50	56.88	84.38	91.88	53.75	1252.59	883.31	708.22
	Y	8	98.00	98.00	93.00	97.50	95.00	92.50	95.00	92.50	85.00	1094.07	821.71	460.30
		16	96.00	93.00	85.00	98.75	92.50	76.25	95.00	85.00	67.50	1135.61	867.24	571.55
		32	93.00	93.00	78.00	81.25	83.12	53.75	75.62	76.88	49.38	1251.20	972.33	674.09
Loop	N	8	100.00	100.00	98.00	100.00	100.00	82.50	100.00	100.00	80.00	1874.00	1570.07	1092.79
		16	100.00	100.00	100.00	100.00	86.25	76.25	100.00	86.25	76.25	1963.12	1601.03	1251.62
		32	98.00	99.00	100.00	100.00	86.25	38.75	97.50	86.25	38.75	1936.27	1818.59	1598.19
	Y	8	95.00	88.00	78.00	100.00	60.00	92.50	95.00	55.00	70.00	1894.33	4554.25	1310.58
		16	96.00	91.00	90.00	90.00	28.75	66.25	88.75	22.50	57.50	1969.37	4481.38	1378.32
		32	89.00	91.00	88.00	80.62	13.75	31.25	76.25	8.12	24.38	2138.35	4274.29	1672.91
Sequence	N	8	98.00	70.00	98.00	97.50	100.00	100.00	95.00	70.00	97.50	1246.43	1298.50	637.11
		16	95.00	70.00	89.00	96.25	100.00	90.00	92.50	70.00	85.00	1188.14	1577.48	785.97
		32	89.00	70.00	84.00	76.25	77.50	64.38	66.88	63.75	59.38	1277.91	1715.05	1013.90
	Y	8	95.00	62.00	80.00	100.00	92.50	95.00	95.00	57.50	80.00	1572.68	1537.60	671.38
		16	89.00	60.00	75.00	86.25	75.00	81.25	78.75	50.00	70.00	1175.86	1640.81	839.37
		32	78.00	62.00	68.00	77.50	53.75	53.75	61.25	39.38	41.88	1293.39	1802.09	1042.54

Table 6: Performance of different planner modules with the scalability in the number of agents (N) and specification complexity for the DubinsCar Environment with obstacles.

Environment. The average improvement in success rate of 11% for our GNN-ODE planner over the MILP planner is similar to the SingleIntegrator environment (Table 5) due to GCBF+controller being more effective at collision avoidance with the linear dynamics of the DoubleIntegrator environment.

C.4 Real-world Drone Experiments

The experimental validation of this methodology involved deploying a fleet of 5 DJI Tello Ryze drones to track the trajectories generated via the Dubins Car model. The drones were configured in WiFi mode to enable swarm behavior which was facilitated through the open-source DJITelloPy² library.

²<https://github.com/damiafuentes/DJITelloPy>

Metric Planner			Finish Rate \uparrow			Safety Rate \uparrow			Success Rate \uparrow			TIR \downarrow		
Spec	Obs	N	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY	GNN-ODE	ODE	STLPY
Branch	N	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	1384.75	536.00	379.50
		16	100.00	100.00	91.00	100.00	100.00	100.00	100.00	100.00	91.25	1768.38	1577.50	474.44
		32	99.00	91.00	73.00	100.00	100.00	100.00	99.38	90.62	72.50	2510.85	2206.44	617.02
	Y	8	100.00	100.00	98.00	97.50	100.00	100.00	97.50	100.00	97.50	2774.50	533.50	390.57
		16	99.00	99.00	94.00	98.75	98.75	100.00	97.50	97.50	93.75	2923.95	1594.28	533.49
		32	99.00	91.00	68.00	96.25	93.75	98.12	95.00	85.62	67.50	2543.79	2263.78	666.90
Cover	N	8	100.00	100.00	93.00	100.00	100.00	100.00	100.00	100.00	92.50	1681.00	738.00	572.20
		16	100.00	100.00	89.00	100.00	100.00	100.00	100.00	100.00	88.75	2127.43	894.50	645.93
		32	96.00	75.00	76.00	100.00	100.00	100.00	95.62	75.00	76.25	2201.11	1542.29	877.32
	Y	8	100.00	100.00	90.00	100.00	100.00	97.50	100.00	100.00	87.50	1649.50	767.50	498.07
		16	99.00	100.00	89.00	98.75	98.75	100.00	97.50	98.75	88.75	2123.30	926.75	756.73
		32	95.00	79.00	78.00	95.00	96.25	98.75	90.62	76.25	77.50	1630.34	1625.60	949.62
Loop	N	8	95.00	98.00	100.00	100.00	100.00	100.00	95.00	97.50	100.00	1951.71	2716.54	1219.75
		16	98.00	99.00	100.00	100.00	100.00	100.00	97.50	98.75	100.00	2612.70	3273.04	1781.62
		32	98.00	93.00	96.00	100.00	100.00	100.00	98.12	93.12	96.25	3271.47	5260.64	2703.45
	Y	8	95.00	98.00	100.00	100.00	95.00	100.00	95.00	92.50	100.00	2533.32	2785.64	1229.75
		16	96.00	99.00	98.00	100.00	96.25	97.50	96.25	95.00	95.00	2713.48	3431.33	1830.21
		32	98.00	93.00	97.00	98.75	80.62	93.75	96.25	75.62	90.62	3353.94	5251.58	2850.15
Sequence	N	8	100.00	80.00	85.00	100.00	100.00	100.00	100.00	80.00	85.00	1565.50	1162.35	693.67
		16	99.00	88.00	70.00	100.00	100.00	100.00	98.75	87.50	70.00	1667.12	1472.54	926.60
		32	81.00	49.00	28.00	100.00	100.00	100.00	80.62	48.75	28.12	1929.73	1979.83	912.03
	Y	8	100.00	88.00	80.00	100.00	100.00	100.00	100.00	87.50	80.00	1552.75	1380.57	846.60
		16	100.00	84.00	60.00	100.00	98.75	97.50	100.00	82.50	60.00	1684.75	1574.84	960.90
		32	84.00	51.00	34.00	99.38	93.12	93.75	83.12	49.38	33.12	1969.68	2003.38	930.35

Table 7: Performance of different planner modules with the scalability in the number of agents (N) and specification complexity for the DoubleIntegrator Environment.

Each Tello drone is equipped with an Inertial Measurement Unit (IMU), a forward-facing camera, and a downward-facing camera. The latter is useful for precise hovering and position estimation using the Vision Positioning System (VPS). However, this system is inaccurate and unreliable as the drones do not possess other sensors like lidar or depth cameras. To mitigate drift and correct the position estimate errors, ArUco tags were utilized to make the trajectory following robust for each drone. This ensured the swarm of drones could accurately follow the designated trajectory as evidenced in the simulation results.

D Hardness of MA-STL specifications

We also performed an ablation study (Table 8, 9, 10) to emphasize the challenge of satisfying these individual temporal objectives while ensuring global constraints such as safety (collision avoidance). For each specification, we use the MILP planner with GCBF+ controller and consider the case of an algorithm prioritizing safety above all else while sacrificing objective satisfaction (i.e. by attempting to remain stationary rather than risking collisions when a collision is detected 1 step ahead). This method (marked ‘Prioritize Safety’) uses environment dynamics and global agent communication to perform this one step lookahead. Notably this is not guaranteed to be safe due to agent input limits and is unrealistic since it is not decentralized. We compare this to an algorithm with the MILP planner that can satisfy the temporal specifications nearly always yet allows collisions between agents by simply following the nominal controller (PID with no collision avoidance maneuvering [21]) marked ‘Prioritize Objective’.

From the results in Table 2 we can see the non-linear nature of the DubinsCar environment, and input limits, hinders a near 100% safety rate in the ‘Prioritize Safety’ variant unlike in the other linear environments (Tables 8, 10). Additionally based on the results we can see in the more complex environments (DubinsCar, DoubleIntegrator) the overly conservative approach of ‘Prioritize Safety’ affects finish rates negatively in the $N = 32$ case (even with the expensive global communication). These results highlight the need for planning informed of collision avoidance procedures.

Planner		Prioritize Objective				Prioritize Safety			
Spec	N	Finish Rate ↑	Safety Rate ↑	Success Rate ↑	TtR ↓	Finish Rate ↑	Safety Rate ↑	Success Rate ↑	TtR ↓
Branch	8	100.00	20.00	20.00	233.25	100.00	100.00	100.00	256.00
	16	100.00	2.50	2.50	236.12	100.00	100.00	100.00	290.50
	32	100.00	0.00	0.00	234.88	100.00	95.00	95.00	342.44
Cover	8	100.00	5.00	5.00	309.50	100.00	100.00	100.00	350.00
	16	100.00	0.00	0.00	310.25	100.00	100.00	100.00	386.00
	32	100.00	0.00	0.00	310.12	100.00	100.00	100.00	471.50
Loop	8	100.00	0.00	0.00	654.75	100.00	100.00	100.00	751.00
	16	100.00	0.00	0.00	654.75	100.00	100.00	100.00	855.38
	32	100.00	0.00	0.00	653.50	100.00	100.00	100.00	1053.81
Sequence	8	100.00	0.00	0.00	419.50	100.00	100.00	100.00	468.00
	16	100.00	0.00	0.00	422.75	100.00	100.00	100.00	563.75
	32	100.00	0.00	0.00	421.62	100.00	98.75	98.75	748.12

Table 8: Depicting the balance between performance and safety with regards to STL specification complexity for the SingleIntegrator Environment at various scales.

Planner		Prioritize Objective				Prioritize Safety			
Spec	N	Finish Rate ↑	Safety Rate ↑	Success Rate ↑	TtR ↓	Finish Rate ↑	Safety Rate ↑	Success Rate ↑	TtR ↓
Branch	8	100.00	25.00	25.00	330.50	100.00	100.00	100.00	370.50
	16	100.00	3.75	3.75	335.12	98.00	92.50	92.50	434.55
	32	100.00	0.62	0.62	332.25	65.00	56.25	47.50	540.42
Cover	8	100.00	5.00	5.00	447.50	100.00	95.00	95.00	483.00
	16	100.00	0.00	0.00	447.50	100.00	98.75	98.75	588.00
	32	100.00	0.00	0.00	447.00	80.00	72.50	61.25	704.72
Loop	8	100.00	0.00	0.00	1006.00	95.00	90.00	85.00	1087.79
	16	100.00	0.00	0.00	1009.12	98.00	77.50	76.25	1568.79
	32	100.00	0.00	0.00	1006.31	79.00	51.25	44.38	1810.24
Sequence	8	100.00	0.00	0.00	582.50	100.00	100.00	100.00	682.00
	16	100.00	0.00	0.00	588.50	95.00	96.25	93.75	814.29
	32	100.00	0.00	0.00	587.12	81.00	71.25	62.50	992.41

Table 9: Depicting the balance between performance and safety with regards to STL specification complexity for the DubinsCar Environment at various scales.

Planner		Prioritize Objective				Prioritize Safety			
Spec	N	Finish Rate ↑	Safety Rate ↑	Success Rate ↑	TtR ↓	Finish Rate ↑	Safety Rate ↑	Success Rate ↑	TtR ↓
Branch	8	100.00	7.50	7.50	316.25	100.00	100.00	100.00	427.50
	16	100.00	3.75	3.75	321.38	91.00	100.00	91.25	547.60
	32	100.00	0.62	0.62	319.06	71.00	100.00	71.25	602.46
Cover	8	100.00	7.50	7.50	404.50	100.00	100.00	100.00	520.00
	16	100.00	0.00	0.00	404.75	98.00	100.00	97.50	745.25
	32	100.00	0.62	0.62	404.50	86.00	100.00	86.25	885.43
Loop	8	100.00	0.00	0.00	812.25	100.00	100.00	100.00	1207.25
	16	100.00	0.00	0.00	812.88	100.00	100.00	100.00	1800.38
	32	100.00	0.00	0.00	814.44	95.00	98.75	95.00	2861.72
Sequence	8	100.00	2.50	2.50	535.00	100.00	100.00	100.00	770.00
	16	100.00	1.25	1.25	537.75	86.00	100.00	86.25	1009.40
	32	100.00	0.00	0.00	538.12	37.00	100.00	36.88	1048.70

Table 10: Depicting the balance between performance and safety with regards to STL specification complexity for the DoubleIntegrator Environment at various scales.