Guiding Neuro-Symbolic Scenario Generation with Spatio-Temporal Logic

Lorenzo Bonin*
University of Trieste
Trieste, Italy
lorenzo.bonin@phd.units.it

Francesco Giacomarra*
University of Trieste
Trieste, Italy
francesco.giacomarra@phd.units.it

Luca Bortolussi
University of Trieste
Trieste, Italy
lbortolussi@units.it

Jyotirmoy V. Deshmukh
University of Southern California
Los Angeles, California, USA
jyotirmoy.deshmukh@usc.edu

Francesca Cairoli
University of Trieste
Trieste, Italy
francesca.cairoli@units.it

ABSTRACT

The rapid advancement of autonomous driving (AD) technologies has outpaced the development of robust safety evaluation methods. Conventional testing relies on exposing AD systems to vast numbers of real-world traffic scenes—a brute-force approach that is prohibitively expensive and statistically ineffective at capturing the rare, safety-critical edge cases essential for validating real-world robustness. To address this fundamental limitation, we introduce *STRELGen*, a scalable framework for the targeted generation of safety-critical driving scenarios. *STRELGen* synergistically combines a multi-agent trajectory-generation diffusion model (DM) with Spatio-Temporal Logic (STREL) specifications that encode complex safety and realism properties through a highly interpretable formalism. Crucially, monitoring satisfaction levels of these specifications is differentiable, enabling gradient-based search. At inference time, we optimize directly over the DM's latent space to maximize STREL formula satisfaction. The result is efficient generation of highly plausible yet safety-critical multi-agent scenarios that lie within the learned data distribution. *STRELGen* thus provides a flexible, interpretable, and powerful tool for stress-testing autonomous driving systems, moving beyond the limitations of brute-force data collection.

KEYWORDS

Scenario Generation; Autonomous Driving; Deep Generative Models; Spatio-Temporal Logic

ACM Reference Format:
Lorenzo Bonin, Francesco Giacomarra, Luca Bortolussi, Jyotirmoy V. Deshmukh, and Francesca Cairoli. 2026. Guiding Neuro-Symbolic Scenario Generation with Spatio-Temporal Logic. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/JCRA2597

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/JCRA2597

1 INTRODUCTION

The integration of autonomous vehicles (AVs) into everyday transportation is progressing rapidly, but this trend is accompanied by growing concerns regarding safety. Developing self-driving systems capable of ensuring reliable and secure operation represents a multifaceted and technically demanding challenge. Machine learning (ML) has enabled autonomous driving systems to achieve human-competitive reliability and robustness across diverse scenarios, a milestone demonstrated by large-scale deployments like the Waymo fleet [4]. However, designing rigorous and comprehensive testing environments remains a major difficulty, particularly when it comes to exposing the system to rare but safety-critical scenarios essential for robust evaluation. Current approaches rely on large-scale real-world deployments and simulator-based replay of safety-critical events [24]. Nonetheless, the enormous search space, intricate interactions, and rarity of critical events make autonomous vehicle safety testing both highly challenging and often ineffective [11, 20]. Log-based simulators are limited by finite dataset diversity, hindering scalability for reliable training and evaluation.

In general, AV environments can be viewed as multi-agent systems, with each test *scenario* representing a set of physically realistic trajectories within this system. With recent advances in deep generative models, particularly diffusion models (DMs) [7, 18], a promising approach is to generate scenarios directly, rather than relying solely on samples from real-world data [14]. In this context, DMs can be used to generate trajectories for autonomous agents, typically conditioned on contextual information such as road maps, traffic configurations, or environmental settings to ensure consistency with scene dynamics. However, without a dedicated mechanism to control the generation process after the training phase, the objective remains limited to producing realistic trajectories under the given context, without explicitly steering the model toward desired behaviors. Consequently, evaluating the system under specific, potentially rare or challenging behaviors—those unlikely to arise in typical trajectories—remains an open problem. To address this, other works [8, 17, 26] incorporate guidance mechanisms to elicit desired properties. Such approaches achieve controllable generation by introducing guidance at each step of the denoising process of the DM, where network outputs are perturbed using the gradient of a differentiable objective to promote desired behaviors. When applied to traffic simulation, however, designing and implementing such objectives—e.g., adherence to traffic rules,

safety distance, realism—becomes highly challenging due to the inherently spatio-temporal and multi-agent characteristics of the domain. To tackle this challenge, Zhong et al. [27] proposed leveraging the established syntax of Signal Temporal Logic (STL) [12]. As a formal language designed for specifying temporal constraints, STL provides a systematic and scalable framework for defining driving rules. In addition, it incorporates a quantitative notion of robustness, allowing the degree to which rules are satisfied to be formally measured. While STL is well-suited for specifying properties for a single agent or a small, fixed pool, it faces significant scalability challenges in multi-agent systems. The core issue is that specifying the complex interactions between agents exponentially increases the complexity and size of the STL formulae. This combinatorial explosion not only affects computational performance but, more critically, severely compromises the interpretability of the requirements. Consequently, writing exhaustive and correct specifications for collaborative or competitive behaviors becomes an inherently hard and error-prone task.

In this work, we extend controllable trajectory generation by leveraging Signal Spatio-Temporal Reach and Escape Logic (STREL) [1, 13], which generalizes STL by incorporating explicit spatial relationships between agents and the environment. At any given timestep, the scene is modeled as a graph where nodes represent agents (e.g., cars, pedestrians). Each node is characterized by a set of attributes, including categorical features like agent type and continuous states like position and velocity. Compared to STL, STREL enables more expressive specifications that capture both temporal and spatial dependencies, allowing fine-grained control over realistic traffic scenarios. A significant limitation of STREL is its difficulty in monitoring the quantitative satisfaction of properties involving categorical predicates. To overcome this, we introduce Colored STREL, an extension of STREL that enables the specification of predicates over specific node types.

Our approach partitions the spatial graph into color-specific subgraphs, where colors represent categorical attributes (e.g., vehicle type in autonomous driving scenes). This allows Colored STREL to monitor spatial properties for particular agent classes and, crucially, to derive continuous robustness values even from discrete categorical attributes. This capability facilitates the precise enforcement of complex, interpretable behaviors in generated trajectories.

Furthermore, we leverage Colored STREL for guided trajectory generation. By formulating specifications as objective functions, we can steer a generative model toward high-satisfaction outputs. This guidance is fully differentiable, enabling efficient gradient-based optimization to identify latent inputs that produce trajectories satisfying our requirements, all without the need for model retraining. Overall, our approach enables controllable trajectory generation, ensuring that the generated scenarios adhere to clearly defined, semantically meaningful specifications.

The *main contributions* presented in this paper can be summarized as follows:

1. **Colored STREL: A Colored Spatio-Temporal Logic.** We introduce Colored STREL, a novel extension of STREL that partitions spatial graphs by node attributes (e.g., vehicle type). This enables the monitoring of complex spatial properties across specific classes of objects while maintaining continuous robustness semantics. The resulting monitor is fully differentiable.
2. **Colored STREL Guidance.** We present a differentiable guidance framework that uses Colored STREL specifications as objective functions. This allows us to directly optimize the latent space of generative models to produce trajectories that satisfy spatio-temporal requirements.
3. **Safety-Critical Data Augmentation.** We present a data augmentation strategy that uses our guidance framework to generate diverse, realistic, and safety-critical scenarios that are underrepresented in real-world data, ensuring consistency with traffic rules and preventing unrealistic outputs.

Related Work

Scenario Generation with Diffusion Models. Several works have leveraged diffusion models to synthesize adversarial or risky driving behaviors targeting the ego vehicle. Xu et al. [26] proposed DiffScene, where a diffusion model is first trained to generate goal-agnostic trajectories of surrounding vehicles. At inference time, adversarial guidance is applied by optimizing a composite risk objective, which balances safety (increasing ego-vehicle risk), functionality (maintaining task feasibility), and realism. While effective, their experiments primarily focus on single surrounding vehicles, limiting scalability to complex multi-agent settings.

Other approaches emphasize the generation of realistic multi-agent interactions. Pronovost et al. [14] introduced a latent diffusion model conditioned on map-based Bird's-Eye-View representations and structured token descriptions to generate realistic poses and trajectories of multiple agents in the Argoverse 2 dataset. Huang et al. [8] proposed Versatile Behavior Diffusion, which integrates Transformer-based encoders and denoisers with classifier guidance to generate joint behaviors of multiple traffic agents. Rowe et al. [16] presented Scenario Dreamer, a vectorized latent diffusion approach that separately generates initial traffic scenes and closed-loop behaviors, and supports exponential tilting to bias scenarios towards adversarial or benign outcomes. These works highlight the growing interest in scaling diffusion-based scenario generation to complex, interactive traffic environments.

Temporal Logic for Autonomous Driving. A line of research has explored the integration of Signal Temporal Logic (STL) into the autonomous driving domain. The techniques in [2, 3, 6, 10] explicitly predict traffic rule violations by combining temporal logic with neural networks. In these frameworks, traffic rules are formalized in STL, and robustness values are computed to serve as features for learning models that anticipate violations in highway driving scenarios. Results show that these predictive monitoring approaches can outperform conventional methods that first predict trajectories and then check for compliance, highlighting the potential of temporal logic as a supervisory signal.

In parallel, NVIDIA researchers introduced a framework that uses STL-based guidance to generate and evaluate safety-critical driving scenarios [27]. Their philosophy is similar to ours in that STL formulae are used to encode safety properties and realism constraints for autonomous vehicle testing. However, their approach relies on gradient-guided optimization at generation time, whereas our method performs latent-space search using STREL, which is

more well-fitted for spatial domains. This distinction is crucial, as it allows us not only to target safety-critical scenarios but also to systematically assess the coverage of the generative model with respect to rare events, something not addressed in prior works.

Noise-Optimization Guidance Approaches. Related to our proposed guidance approach, there exists a growing line of works that explore the possibility of steering the generation process from a pre-trained diffusion model by optimizing the noise vector in the latent space, according to some differentiable objective function defined on the output space. DOODL [22] and Direct Noise Optimization (DNO) [9] exemplify this trend, as they both present a guidance approach based on back-propagating through the entire reverse process to find latent points that yield outputs best aligned with a target classifier or motion objective. A similar approach to what we propose can be found in [5], where robustness with respect to STL formulae is used to guide an optimization process in the latent space of score-based diffusion models.

Such methods enable fine-grained, differentiable control over generation without retraining, but require careful regularization to avoid drifting away from the data manifold.

2 BACKGROUND

In this section, we review the theoretical foundations underlying our approach.

2.1 Diffusion Models

Diffusion probabilistic models [18] are generative models that learn data distributions by gradually adding and then removing noise from training samples. We consider the denoising diffusion probabilistic model (DDPM) [7], which can be viewed as a discretized score-based model trained with Langevin dynamics [21].

Forward and Reverse Processes. The forward process progressively perturbs a clean sample x^0 through a sequence of $\mathcal{T}$ injections of Gaussian noise with variances β_τ, $\tau \in \{1, \dots, \mathcal{T}\}$, until it reaches a pure noise distribution $x^{\mathcal{T}} \sim \mathcal{N}(\vec{0}, I)$. The reverse process, parameterized by a neural denoiser $\epsilon_\theta(x^\tau, \tau)$, learns to invert this corruption, reconstructing clean samples from noisy ones. The model is trained via a simple denoising objective that minimizes the prediction error between true and estimated noise:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{\tau, x_0}\left[\|\epsilon - \epsilon_\theta(x^\tau, \tau)\|^2\right], \tag{1}$$

where $\tau \sim \mathit{Unif}(\{1, \dots, \mathcal{T}\})$ and x_0 is sampled from the dataset. The model can then generate new samples by iteratively denoising pure noise drawn from $\mathcal{N}(\vec{0}, I)$.

Conditional Diffusion. To guide generation, diffusion models can be conditioned on auxiliary information y (e.g., scene context or class labels), learning the conditional distribution $p(x^0 \mid y)$. The same denoising loss applies, with the denoiser now depending on y, i.e., $\epsilon_\theta(x^\tau, \tau \mid y)$. This enables targeted generation of trajectories consistent with external constraints.

Implicit and Latent Variants. The sampling process in DDPMs is slow, as it requires executing a long chain of probabilistic denoising steps. Denoising Diffusion Implicit Models (DDIMs) [19] allow for faster sampling by removing stochasticity from all denoising steps except the last one (step $\mathcal{T}$). Therefore, given a sample $x^{\mathcal{T}} \sim \mathcal{N}(\vec{0}, I)$, the rest of the denoising process becomes deterministic. The sampling process can be then accelerated by using only a subset of the diffusion steps. Finally, Latent Diffusion Models (LDMs) [15] perform the diffusion process in a compressed latent space learned by an autoencoder, greatly improving efficiency while maintaining generative performances.

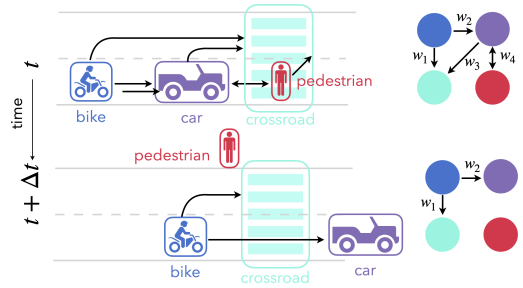

Figure 1: Example of a safety-critical AD scene represented as a STREL dynamic graph: the motorbike sees the crossroad but not the pedestrian, as it is hidden by the car in front.

2.2 Spatio-Temporal Logic

Spatio-Temporal Reach and Escape Logic (STREL) [1, 13] is a formal language designed to express complex, spatio-temporal relationships between interacting agents. In this framework, a multi-agent system is modeled as a dynamic graph $G(t) = (L(t), E(t))$, where $L(t)$ is the set of agents (nodes) and $E(t)$ represents their possible interactions (edges) at time t. Each edge has two key properties: a binary weight indicating whether two agents are interacting (e.g., if they see each other), and a distance metric f that quantifies a specific notion of proximity between them. This metric f is a key source of STREL's expressivity, allowing it to define a wide range of spatial properties. Crucially, the graph is dynamic; the set of agents, their attributes, and the connections between them all evolve over time, as illustrated in Fig. 1.

STREL properties are defined by the following *syntax*:

$$\varphi := \mathit{true} \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathsf{U}_{[t_1,t_2]} \varphi_2 \mid \varphi_1 \mathsf{R}^f_{[d_1,d_2]} \varphi_2 \mid \mathsf{E}^f_{[d_1,d_2]} \varphi$$

where μ denotes the atomic predicates, $\neg$, $\wedge$ denote negation and disjunction. On one hand we have the temporal operator *Until*, $\varphi_1 \mathsf{U}_{[t_1,t_2]} \varphi_2$, that denotes that φ_1 is satisfied until, in a time between t_1 and t_2 time units in the future, φ_2 becomes true. Temporal operators are evaluated in each location separately. On the other hand, $\mathsf{R}^f_{[d_1,d_2]} \varphi_2$ and $\mathsf{E}^f_{[d_1,d_2]} \varphi$ denote the spatial operators of *Reach* and *Escape*, with $f : L \times L \to B$ a distance function (B the distance domain). The reachability operator describes the property of reaching a location satisfying φ_2, through a path with all locations that satisfy φ_1 with path length between d_1 and d_2. The escape operator describes the possibility of escaping from a certain region via a path passing only through locations that satisfy φ, with the distance between the starting location of the path and the last (not the path length) that belongs to the interval $[d_1, d_2]$. Spatial operators are evaluated at each time step separately. From this essential syntax, we can define as usual other operators as follows:

$false := \neg true$, $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\mathsf{F}_{[t_1,t_2]}\varphi := true\ \mathsf{U}_{[t_1,t_2]}\varphi$ and $\mathsf{G}_{[t_1,t_2]}\varphi := \neg\mathsf{F}_{[t_1,t_2]}\neg\varphi$, where $\mathsf{F}_{[t_1,t_2]}\varphi$ and $\mathsf{G}_{[t_1,t_2]}\varphi$ denote respectively the *Eventually* (φ will hold at some point in the time interval $[t_1, t_2]$) and *Globally* (φ holds at all times in the interval $[t_1, t_2]$) operator.

We can also derive other three spatial operators: **somewhere**: $\mathsf{SW}^f_{[0,d]}\varphi := true\mathsf{R}^f_{[0,d]}\varphi$, **everywhere**: $\mathsf{EW}^f_{[0,d]}\varphi := \neg\mathsf{SW}^f_{[0,d]}\neg\varphi$ and **surround**: $\varphi_1\mathsf{Surr}^f_{[0,d]}\varphi_2 := \varphi_1 \wedge \neg\big(\varphi_1\mathsf{R}_{[0,d]}\neg(\varphi_1 \vee \varphi_2)\big) \wedge \neg(\mathsf{E}_{[d,\infty]}\varphi_1)$. *Somewhere* and *Everywhere* operators describe behaviour for some or all locations within reach of a specific location, *Surround* expresses the notion of being surrounded by a region that satisfies φ_2, while being in a φ_1 satisfying region.

STREL Semantics. The semantics of STREL is evaluated pointwise at each time t and at each location ℓ. Let G be a spatial model (i.e. a graph with a time-varying edge relation) with L the space universe (i.e. the set of locations), D_1 and D_2 be two signal domains, and $\vec{x}$ be a spatio-temporal D_1-trace for locations in L. The D_2-monitoring function $\mathbf{m}$ of $\vec{x}$ is recursively defined as follows.

$$\mathbf{m}(G,\vec{x},\mu,t,\ell) = g(\mu,\vec{x}(\ell,t))$$

$$\mathbf{m}(G,\vec{x},\neg\varphi,t,\ell) = \odot\mathbf{m}(G,\vec{x},\varphi,t,\ell)$$

$$\mathbf{m}(G,\vec{x},\varphi_1\wedge\varphi_2,t,\ell) = \mathbf{m}(G,\vec{x},\varphi_1,t,\ell)\otimes\mathbf{m}(G,\vec{x},\varphi_2,t,\ell)$$

$$\mathbf{m}(G,\vec{x},\varphi_1\mathsf{U}_{[t_1,t_2]}\varphi_2,t,\ell) = \bigoplus_{t'\in[t+t_1,t+t_2]}\Big[\mathbf{m}(G,\vec{x},\varphi_2,t',\ell)\otimes\Big(\bigotimes_{t"\in[t,t']}\mathbf{m}(G,\vec{x},\varphi_2,t",\ell)\Big)\Big]$$

$$\mathbf{m}(G,\vec{x},\varphi_1\mathsf{R}^f_{[d_1,d_2]}\varphi_2,t,\ell) = \bigoplus_{\tau\in Routes(G(t),\ell)}\Big[\bigoplus_{i:\left(d^f_\tau[i]\in[d_1,d_2]\right)}\Big(\mathbf{m}(G,\vec{x},\varphi_2,t,\tau[i])\otimes\bigotimes_{j<i}\mathbf{m}(G,\vec{x},\varphi_1,t,\tau[j])\Big)\Big]$$

$$\mathbf{m}(G,\vec{x},\mathsf{E}^f_{[d_1,d_2]}\varphi,t,\ell) = \bigoplus_{\tau\in Routes(G(t),\ell)}\Big[\bigoplus_{\ell'\in\tau:\left(d^f_{G(t)}[\ell,\ell']\in[d_1,d_2]\right)}\Big(\bigotimes_{i\le\tau(\ell')}\mathbf{m}(G,\vec{x},\varphi,t,\tau[i])\Big)\Big]$$

Here, $Routes(G(t),\ell)$ denotes the set of routes in $G(t)$ starting from $\ell \in L$. To ease the notation we use $\otimes, \oplus, \odot$ to denote respectively conjunction $\otimes_{D_2}$, disjunction $\oplus_{D_2}$ and negation $\odot_{D_2}$. For the Boolean signal domain ($D_2 = \{\bot, \top\}$), we say that $(G, \vec{x}(\ell, t))$ satisfies a formula φ iff $\mathbf{m}(G,\vec{x},\varphi,t,\ell) = \top$. For max/min signal domain ($D_2 = \mathbb{R}^\infty$) we say that $(G, \vec{x}(\ell, t))$ satisfies a formula φ iff $\mathbf{m}(G,\vec{x},\varphi,t,\ell) > 0$.

Example 1. *Fig. 1 illustrates the evolution in time of a safety-critical autonomous driving scene and its corresponding abstraction as STREL dynamic graph. In this model, a directed edge from node a to node b exists if and only if agent a perceives agent b. Crucially, the graph structure depends on the chosen distance function f, as different functions can model different perceptual or spatial relationships. In our example, the function emulates a LiDAR such that the motorbike perceives the crosswalk and the car in front, but not the occluded pedestrian behind it, accurately reflecting the scene's visibility constraints. A potential safety-critical situation occurs if the motorbike finds a pedestrian ahead within the safety distance d_{safe}. This is captured by the following STREL requirement:*

$$\varphi = \mathsf{F}_{[0,T]}\Big[\big(\texttt{isMotorBike}(\vec{x})\big)\mathsf{R}^{Front}_{[0,d_{safe}]}\big(\texttt{isPedestrian}(\vec{x})\big)\Big], \quad (2)$$

where $\texttt{isMotorBike}(\vec{x})$ *and* $\texttt{isPedestrian}(\vec{x})$ *are two atomic predicates denoting whether the node is of type motorbike or pedestrian.*

This very simple example highlights a big *limitation of STREL*. We are only able to monitor the Boolean satisfaction of φ, whereas the quantitative semantics is not well-defined over categorical attributes; we are not able to quantify how close the scene is to a safety violation. Therefore, the satisfaction of STREL formulae is unsuitable as an objective function for optimization problems involving categorical data, such as vehicle type.

3 NEUROSYMBOLIC SCENARIO GENERATION

In this section, we present our *STRELGen* approach for autonomous driving scenario generation. The primary objective is to enable controllable generation of multi-agent trajectories that are both realistic and consistent with semantically meaningful specifications. To this end, we combine latent DMs for trajectory synthesis with CSTREL-based guidance that steers the generative process toward desired behaviors.

Our method introduces two key innovations. First, we extend STREL with a coloring mechanism, yielding *Colored STREL* (CSTREL), which partitions the spatial graph into subgraphs corresponding to agent types or contextual categories. This extension enables rules to be specified over heterogeneous entities (e.g., cars, pedestrians, cyclists) and provides continuous robustness measures even for discrete categorical attributes. Second, we develop a differentiable guidance mechanism that directly leverages CSTREL robustness values as objectives for trajectory generation. By expressing logical constraints as differentiable functions, this mechanism enables gradient-based search in the latent space, ensuring that generated trajectories satisfy the specified constraints.

3.1 Trajectories Generation

In this work, we adopt the diffusion-based architecture proposed in [23] as our baseline for multi-agent scenario synthesis. A latent diffusion model is used to generate future trajectories of all agents in a traffic scene jointly, while scene context—comprising map features, agent history, and inter-agent interactions—is encoded by a pretrained QCNet [28] and injected into the diffusion model through cross-attention layers. This conditioning ensures that the generated trajectories remain consistent with both scene geometry and agent dynamics. A central contribution of [23] is the introduction of Optimal Gaussian Diffusion (OGD), which replaces the standard Gaussian prior with a data-dependent "optimal" prior derived from statistics of marginal trajectories at a small diffusion time step. This design allows the model to achieve high-quality predictions with substantially fewer denoising steps.

In our framework, we adopt the implicit formulation of [23], which accelerates sampling and, crucially, renders the reverse process deterministic. This property is particularly important for our method, as it enables stable backpropagation of gradient-based CSTREL specifications defined on generated trajectories with respect to latent inputs.

3.2 Colored STREL

Scenarios can be naturally interpreted as networks of interacting agents. Spatio-temporal logic (STREL in Section 2.2) is tailored to monitor dynamic networks of spatially-distributed agents. A key feature is the ability to automatically monitor both Boolean and quantitative satisfaction of spatial and temporal properties, providing insights into complex behaviors that emerge from local and dynamic interactions. However, the STREL quantitative semantics—a core metric in our guidance framework—is currently ill-defined for properties containing categorical predicates (e.g., `isBike` or `isPedestrian`), as shown in Example 1. To address this, we introduce a colored formulation, Colored STREL (CSTREL), which enables the specification and quantitative verification of type-specific requirements. Given the set of all colors C, e.g. $C = \{\texttt{car}, \texttt{bike}, \texttt{pedestrian}, \texttt{traffic light}, \texttt{crossroad}\}$, the syntax of Colored STREL is:

$$true \mid \mu^c \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathsf{U}_{[t_1,t_2]} \varphi_2 \mid \varphi_1^{c_1} \mathsf{R}^f_{[d_1,d_2]} \varphi_2^{c_2} \mid \mathsf{E}^f_{[d_1,d_2]} \varphi^c,$$

where $c, c_1, c_2 \subseteq C$ are sets of colors. If $c = c_1 = c_2 = C$, CSTREL reduces exactly to STREL. To better understand the main differences in the colored semantics, let's introduce a *coloring function*, $p : L \times \mathbb{R}^+ \to C$, mapping every pair (ℓ, t), node $\ell \in L$ at time $t \in \mathbb{R}^+$, into the associated color $p(\ell, t) \in C$.

Let $\bot_{D_2}$ denote the minimum satisfaction value, i.e. *False* in the Boolean case and $-\infty$ in the quantitative case. The main differences in CSTREL semantics can be summarized as follows.

Colored Atoms: $\mathbf{m}(S, \vec{x}, \mu^c, t, \ell) = g_c(\mu^c, \vec{x}(\ell, t))$,

where $g_c(\mu, \vec{x}(\ell, t)) = g(\mu, \vec{x}(\ell, t))$ if $p(\ell, t) \in c$, i.e., if node ℓ at time t has a color present in set c, and $g_c(\mu, \vec{x}(\ell, t)) = \bot_{D_2}$ otherwise.

For the colored reach and escape operators, we must address the path-generating function in order to search only over paths with allowed coloring. A collateral advantage of CSTREL is its reduced computational cost, as we considerably reduce the search space.

Colored Reach:
$$\mathbf{m}(S, \vec{x}, \varphi_1^{c_1} \mathsf{R}^f_{[d_1,d_2]} \varphi_2^{c_2}, t, \ell) = \bigoplus_{\tau \in CRoutes(S(t), \ell, c_1, c_2)} \Big[\bigoplus_{i:\left(d^f_\tau[i] \in [d_1,d_2]\right)} \Big(\mathbf{m}(S, \vec{x}, \varphi_2^{c_2}, t, \tau[i]) \otimes \bigotimes_{j<i} \mathbf{m}(S, \vec{x}, \varphi_1^{c_1}, t, \tau[j]) \Big) \Big]$$

Colored Escape:
$$\mathbf{m}(S, \vec{x}, \mathsf{E}^f_{[d_1,d_2]} \varphi^c, t, \ell) = \bigoplus_{\tau \in CRoutes(S(t), \ell, c)} \Big[\bigoplus_{\ell' \in \tau:\left(d^f_{S(t)}[\ell,\ell'] \in [d_1,d_2]\right)} \Big(\bigotimes_{i \le \tau(\ell')} \mathbf{m}(S, \vec{x}, \varphi^c, t, \tau[i]) \Big) \Big]$$

where $CRoutes(S(t), \ell, c_1, c_2)$ is a message passing function that returns all paths such that every visited location has a color in c_1 and terminates with a color in c_2, whereas $CRoutes(S(t), \ell, c)$ returns all paths such that every visited location is in c. The derived spatial operators inherit the coloring scheme from the reach and escape defined above. More precisely:

Colored somewhere: $\mathsf{SW}^f_{[0,d]} \varphi^c := true^C \mathsf{R}^f_{[0,d]} \varphi^c$;

Colored everywhere: $\mathsf{EW}^f_{[0,d]} \varphi^c := \neg \mathsf{SW}^f_{[0,d]} \neg \varphi^c$;

Colored surround: $\varphi_1^{c_1} \mathsf{Surr}^f_{[0,d]} \varphi_2^{c_2} := \varphi_1^{c_1} \wedge \neg\big(\varphi_1^{c_1} \mathsf{R}_{[0,d]} (\neg(\varphi_1^{c_1} \vee \varphi_2^{c_2}))\big) \wedge \neg(\mathsf{E}_{[d,\infty]} \varphi_1^{c_1})$

Example 2. *The STREL formula presented in Example 1, Eq. (2), describing a scene where a motorbike finds a pedestrian in front within the safety distance, can be rewritten as a CSTREL formula:*

$$\varphi = \mathsf{F}_{[0,T]} \Big[\big(Mov^{[\texttt{motorbike}]}(\vec{x})\big) \mathsf{R}^{Front}_{[0,d_{safe}]} \big(Mov^{[\texttt{pedestrian}]}(\vec{x})\big) \Big], \quad (3)$$

where $\mathbf{v}$ *is the agent velocity and* $Mov^{[\texttt{types}]}(\vec{x}) = (\mathbf{v}^{[\texttt{types}]}(\vec{x}) > 0)$ *is the atomic predicated determining whether agents of type* [types] *are moving. The robustness of this CSTREL formula quantifies the level of satisfaction—or criticality—based on the velocities of the motorbike and pedestrian and the proximity of their trajectories.*

Remark 1. *We developed a fully differentiable PyTorch library for CSTREL specifications that also provides the first differentiable implementation of STREL, enabling STREL-based gradient searches.*

3.3 CSTREL-based Guidance

Our CSTREL library allows for the flexible, scenario-specific definition and evaluation of formulae describing the targeted behavior. Given a CSTREL formula φ and a point $z \in \mathcal{Z}$, the latent space of a pre-trained generative model G_θ, we compute a quantitative notion of *robustness* for the corresponding generated trajectory $\tilde{x}^* = G_\theta(z^*)$ in the data space $\mathcal{X}$. This trajectory $\tilde{x}^*$ is a spatio-temporal signal describing the future evolution of the entire scene (all $n = |L|$ agents), conditioned on past observations.

The CSTREL robustness evaluated at time 0 is generally a tensor of size n. We must therefore find a statistic that effectively summarizes the robustness values in this tensor. In our safety-critical guidance pipeline—where properties describe dangerous behaviors—we aim to find latent inputs that lead to at least one safety violation. The maximum (the softmax for a better gradient flow) of the n robustness values is therefore a suitable metric to maximize. Conversely, for realism properties that all agents must satisfy, the minimum value is the appropriate metric to maximize.

Let $\rho^\varphi(\tilde{x}^*)$ denote this real-valued metric, which quantifies the level of satisfaction of φ for the scene $\tilde{x}$. Consequently, ρ^φ serves as a differentiable objective for identifying latent points in $\mathcal{Z}$ that maximize property satisfaction. This guidance effectively forces either at least one agent to violate a safety property or all agents to behave realistically. The two requirements can be combined.

This CSTREL-based guidance strategy is straightforward to implement but may drive the search toward regions of the latent space that lead to scenes that lie off the data manifold [9, 22], i.e. latent inputs with low probability w.r.t. the latent prior $p(z) = \mathcal{N}(0, I)$. To mitigate this issue, we regularize the objective function of our gradient-ascent procedure. More precisely, we jointly optimize over an objective composed of the robustness ρ^φ and a regularization term proportional to the log-likelihood of the latent input w.r.t the latent prior, which is a standard Gaussian. Mathematically, the regularized objective function is

$$\mathcal{J}(z) = \rho^\varphi(G_\theta(z)) - \lambda \cdot \left(\frac{1}{2}\|z\|_2^2\right), \quad (4)$$

where λ is a tunable trade-off parameter. The search can terminate when we find z^* such that $\rho^\varphi(G_\theta(z^*)) > 0$. Incorporating realism within the guidance properties can further discourage off-manifold behaviors or hallucinated trajectories. The overall pipeline of our method is summarized in Algorithm 1.

Algorithm 1 *STRELGen*

Input: Diffusion Model G_θ, CSTREL formula φ, starting latent point $z_0 \sim N(\vec{0}, I)$, learning rate η, regularization parameter λ.
Output: Optimized latent point z_φ.
for $i = 1 : \text{Max}_{step}$ **do**
  $\mathcal{J}(z) = \rho^\varphi(G_\theta(z)) - \lambda \frac{1}{2}\|z\|_2^2$, ▷ optimization objective
  $z \leftarrow \mathbf{GA}(z, \mathcal{J}(z), \eta)$ ▷ one-step of gradient-ascent
if $\rho^\varphi(G_\theta(z_{\text{Max}_{step}})) > 0$ **then** ▷ formula is satisfied
  $z_\varphi = z_{\text{Max}_{step}}$
else
  draw new sample $z_0^* \sim N(\vec{0}, I)$ and **go to** step 1

4 EXPERIMENTS

This section evaluates the proposed approach and addresses the following research questions:

- ***RQ1*** *Can CSTREL guidance reliably steer the diffusion model to produce trajectories that maximize the satisfaction of target specifications?*
- ***RQ2*** *Can we guide the model toward safety-critical scenarios while remaining on the data manifold, avoiding low-probability regions of the latent space?*
- ***RQ3*** *Can CSTREL-based guidance optimize latent variables while preserving the physical plausibility of generated trajectories?*

We first describe the experimental setup, followed by qualitative and quantitative analyses of the results. Our implementation is available at https://github.com/lorenzobonin/strelgen.

4.1 Experimental Setup

Dataset. We evaluate our approach on the Argoverse 2 Motion Forecasting Dataset [25], a large-scale benchmark for controllable trajectory generation. The dataset comprises over 250,000 driving scenarios collected across six diverse U.S. regions, totaling approximately 763 hours of real-world driving data. Each scenario includes a 5-second observation window followed by a 6-second prediction horizon. We use the official training and validation splits.

Training Details. The latent diffusion model is trained for 64 epochs using the same hyperparameter configuration as in [23], to which we refer for further details. Context embeddings are extracted using the pre-trained QCNet model [28]. We employ the implicit formulation of the diffusion process with 100 denoising steps.

Computational Costs. CSTREL preserves the theoretical complexity of [13] over color-based subgraphs. For simple scenarios, our tensorized GPU implementation reduces the time per gradient ascent step of the iterative algorithm in [13] from approximately 8 s to 0.15 s. In more complex settings, this implementation is crucial to maintaining feasibility and computational efficiency.

4.2 CSTREL Specifications

For evaluation, we selected $n_{\text{scen}} = 3$ representative scenarios from the validation set. Specifically, we first identified the ten scenarios containing the largest number of agents and the ten scenarios exhibiting the greatest diversity of agent types. From their union, we sampled three scenarios for detailed analysis. The generative model does not assume a designated ego vehicle, as all agents' trajectories are generated jointly.

To evaluate the effect of CSTREL-based guidance, we designed logical formulae that intentionally promote safety-critical or adversarial behaviors, enabling the model to explore challenging conditions. The set of colors C we considered corresponds to the type of agents that can populate the scene. In our experiments $C = \{\texttt{car}, \texttt{pedestrian}, \texttt{bike}, \texttt{motorcycle}, \texttt{bus}, \texttt{static}\}$. To improve readability, we group a subset of relevant labels under the set $\texttt{vehicles} = \{\texttt{car}, \texttt{motorcycle}, \texttt{bus}\}$. The considered CSTREL specifications are the following.

Fast Vehicle Reaches a Bike or Pedestrian capturing potential near-collision events involving vulnerable road users:

$$\varphi_{pb\_uns} = \mathsf{F}_{[0,T]}\Big[\big(\text{Fast}^{[\texttt{vehicles}]}(\vec{x})\big)\mathsf{R}^{\text{Euclid}}_{[0,d_{\text{safe}}]}\big(\text{Mov}^{[\texttt{ped,bike}]}(\vec{x})\big)\Big].$$

Fast Vehicle Finds a Slow Vehicle Ahead targeting rear-end collision risks:

$$\varphi_{front} = \mathsf{F}_{[0,T]}\Big[\big(\text{Fast}^{[\texttt{car,bus}]}(\vec{x})\big)\mathsf{R}^{\text{Front}}_{[0,d_{\text{safe}}]}\big(\text{Slow}^{[\texttt{car,bus}]}(\vec{x})\big)\Big].$$

Fast Car Surrounded by Slow Cars describing aggressive driving in dense traffic:

$$\varphi_{surr} = \mathsf{F}_{[0,T]}\Big[\big(\text{Fast}^{[\texttt{car}]}(\vec{x})\big)\mathsf{Surr}^{\text{Euclid}}_{[0,d_{\text{safe}}]}\big(\text{Slow}^{[\texttt{car}]}(\vec{x})\big)\Big].$$

The atomic predicates are defined as: $\text{Mov}^{[\texttt{types}]}(\vec{x}) = (\mathbf{v}^{[\texttt{types}]}(\vec{x}) > 0) \wedge (v_{\text{real}} > \mathbf{v}^{[\texttt{types}]}(\vec{x}))$, $\text{Fast}^{[\texttt{types}]}(\vec{x}) = (\mathbf{v}^{[\texttt{types}]}(\vec{x}) > v_{\text{safe}}) \wedge (v_{\text{real}} > \mathbf{v}^{[\texttt{types}]}(\vec{x}))$ and $\text{Slow}^{[\texttt{types}]}(\vec{x}) = (\mathbf{v}^{[\texttt{types}]}(\vec{x}) < v_{\text{slow}})$, where $\mathbf{v}(\vec{x})$ is the velocity of each agent, v_{safe}, v_{real}, and v_{slow} are velocity thresholds and d_{safe} is the safety distance. These parameters are scenario-dependent (e.g., thresholds differ between urban and highway scenes). Each specification is applied to one of the selected scenarios. To further demonstrate the versatility of CSTREL, we introduce two additional formulas that, although not safety-critical, promote realistic motion and smooth behavior.

No Sudden Changes in Heading Direction:

$$\varphi_{\text{head}} = \mathsf{G}_{[0,T]}\big(\mathbf{h}^{[\texttt{car}]}(\vec{x}) < h_{\text{smooth}}\big),$$

No Overlap Among Moving Vehicles: $\varphi_{\text{ov}} = \mathsf{G}_{[0,T]}$

$$\mathsf{EW}^{\text{Euclid}}_{[0,D]}\Big(\neg\big(\text{Mov}^{[\texttt{vehicle}]}(\vec{x}) \wedge \mathsf{SW}^{\text{Euclid}}_{(0,\,d_{\text{safe}}]}\text{Mov}^{[\texttt{vehicle}]}(\vec{x})\big)\Big),$$

where $\mathbf{h}(\vec{x})$ denotes the instantaneous heading change, and D and d_{safe} are distance hyperparameters.

Baseline. As a baseline, we use the vanilla diffusion model without any CSTREL-based guidance. This comparison isolates the contribution of the proposed method by contrasting guided and unguided generations under identical conditions.

Metrics. Performance is assessed through both qualitative and quantitative analyses. Qualitatively, we inspect generated trajectories to evaluate plausibility, diversity, and interpretability (Fig. 2). Quantitatively, we compute the distribution of minimum pairwise distances D between agents (Fig. 3) and the number of collisions observed in each scenario (Table 4). Together, these metrics provide

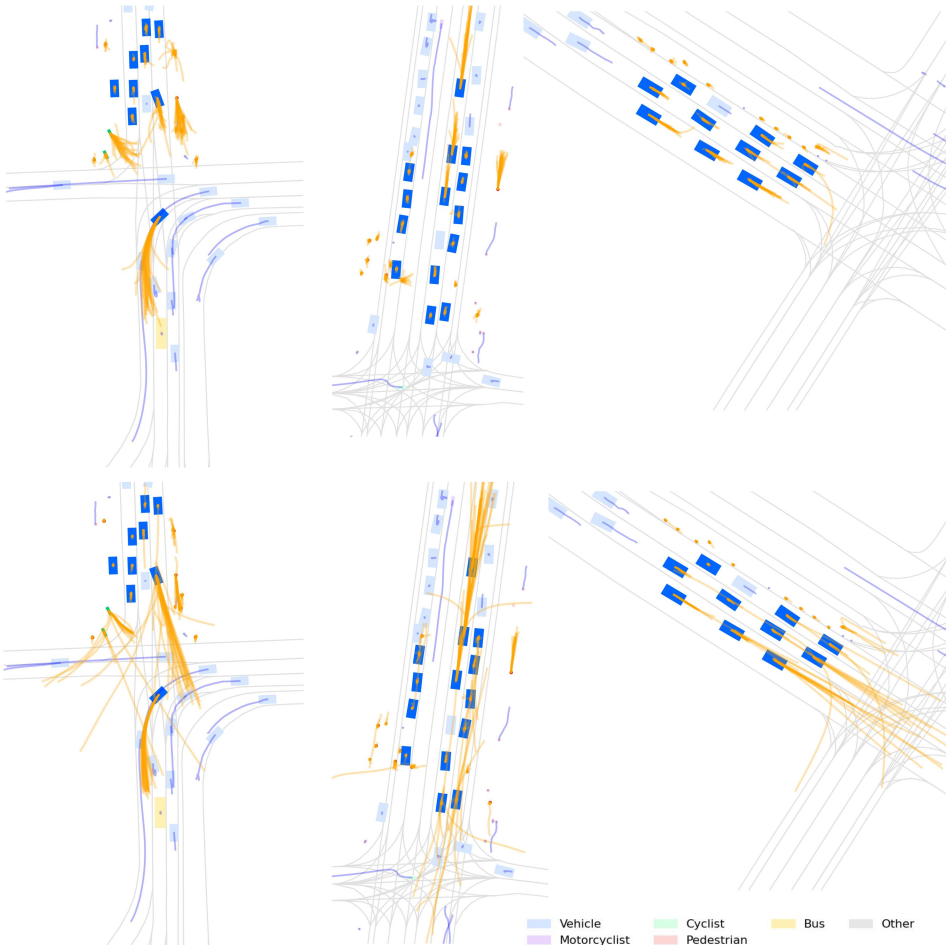

Figure 2: Representative safety-critical scenarios generated with *STRELGen*. Each column corresponds to a target STREL property: (Left) φ_{pb_uns}—*Fast Vehicle Reaches a Bike or Pedestrian*; (Center) φ_{front}—*Fast Vehicle Finds a Slow Vehicle Ahead*; (Right) φ_{surr}—*Fast Car Surrounded by Slow Cars*. Blue trajectories correspond to fixed context agents, and yellow trajectories to optimized predictions. The top row shows unguided (vanilla) samples; the bottom row shows samples optimized via *STRELGen*.

complementary insights into how *STRELGen* steers the diffusion model.

4.3 Results

Fig. 2 shows representative examples of safety-critical scenes generated via *STRELGen*. The top row shows unguided (vanilla) samples, while the bottom row presents optimized trajectories. Across all three properties, the optimized trajectories exhibit behaviors consistent with the target logical formulae: pedestrians and cyclists approach vehicles (**left**); fast vehicles approach slower ones ahead (**center**); and aggressive drivers emerge within congested traffic (**right**). In all cases, trajectories remain spatially coherent and aligned with the road geometry. Importantly, CSTREL ensures that only the agent types specified in each formula exhibit significant behavioral changes relative to the baseline. The likelihood regularization term in Eq. (4) prevents the optimization from deviating toward low-probability regions of the latent space. We also considered composing safety-critical and other realism-oriented formulae (e.g., Eq. (4.2)), but empirically found that using likelihood penalty was sufficient to obtain plausible trajectories.

To quantify the steering effect of *STRELGen*, Fig. 3 reports the distribution of the minimum inter-agent distance D—measured between the center points of the agents—across CSTREL properties, comparing the baseline (blue) and guided (orange) generations. *STRELGen* consistently produces lower median values of D, indicating reduced safety margins and a higher likelihood of unsafe interactions. The method also exhibits larger interquartile ranges, reflecting increased diversity in the generated scenarios. With respect to the considered experiments, the proportion of generated scenarios that are challenging, meaning positive robustness w.r.t. the CSTREL requirements, is around 13,6% under unguided sampling, increasing to 100% with the proposed CSTREL-based guidance strategy. Table 4 reports the number of collisions observed per CSTREL property. We define a potential collision as a vehicle being within 0.9 m of another agent. Across all cases, *STRELGen*

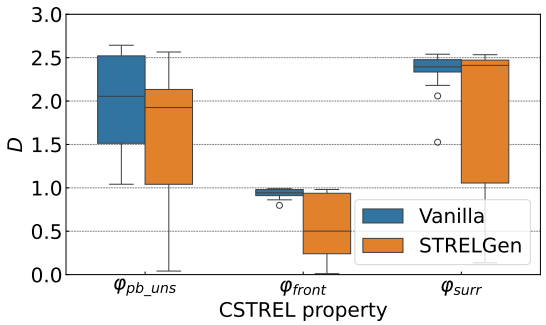

Figure 3: Distribution of minimum inter-agent distance D (in meters) across CSTREL properties; lower values indicate closer agents and thus higher safety risk.

Method / Property	φ_{pb_uns}	φ_{front}	φ_{surr}
Vanilla	0 (0%)	5 (17%)	0 (0%)
STRELGen	5(17%)	19 (63%)	7 (23%)

Figure 4: Potential collision: count and percentage (30 generated scenarios for each method-formula configuration).

consistently increases the number of potential collisions, confirming its effectiveness in generating safety-critical behaviors while remaining within the data manifold.

5 DISCUSSION

In this section, we summarize the key findings that address the research questions posed in this work.

RQ1 *Can CSTREL guidance reliably steer the diffusion model to produce trajectories that maximize the satisfaction of target specifications?*
The results section shows that the proposed approach effectively generates safety-critical scenarios that satisfy the properties specified through CSTREL, offering explicit control over both global scene behavior and agent-specific dynamics. The integration of CSTREL guidance steers the diffusion process toward configurations that maximize the desired specifications, enabling targeted manipulation of safety margins and interaction patterns among agents.

RQ2 *Can we guide the model toward safety-critical scenarios while remaining on the data manifold, avoiding low-probability regions of the latent space?*
Guidance based solely on robustness can push latent points toward the tails of the distribution, producing off-manifold samples. To mitigate this, we incorporated a likelihood penalty into the robustness function. This simple yet effective addition ensures that the optimized latent points generated by *STRELGen* remain consistent with the latent distribution, while achieving high satisfaction values for the target property.

RQ3 *Can CSTREL-based guidance optimize latent variables while preserving the physical plausibility of generated trajectories?*
We explored the inclusion of realism-preserving formulae in the robustness function but found that the likelihood penalty alone was sufficient to obtain plausible trajectories. In fact, our qualitative analysis showed that the generated samples tend to adhere to the underlying physical and environmental constraints. This result indicates that *STRELGen* not only supports targeted scenario generation but also enables analysis of the generative model's coverage with respect to desired properties—revealing its capacity to produce compliant samples while remaining within the data manifold.

Overall, our findings suggest that *STRELGen* can successfully balance specification-driven optimization with the plausibility of generated samples, indicating its potential for controllable and interpretable scenario generation in safety-critical domains.

6 CONCLUSIONS

In this work, we introduced *STRELGen* a controllable trajectory generation framework based on Colored Signal Spatio-Temporal Reach and Escape Logic (Colored STREL), an extension of STREL that enables the specification and quantitative monitoring of spatio-temporal properties across different classes of agents. By leveraging Colored STREL robustness as a differentiable objective, we guide a pre-trained diffusion model in its latent space to generate trajectories that satisfy complex, formally defined safety-critical behaviors.

The proposed guidance strategy, which combines the Colored STREL objective with a likelihood-based regularization term, proved effective in producing safety-critical scenarios. The regularization term ensures that the optimization remains close to the data manifold, while the logical objective directs the generation toward behaviors exhibiting reduced safety margins or other targeted interactions. Overall, these results demonstrate that formal, logic-based guidance can complement data-driven generative models, bridging the gap between realism and controllable scenario synthesis for simulation-based safety testing.

Future work will aim at developing automated pipelines that infer safety-critical CSTREL specifications directly from the underlying road geometry and agent configurations, thereby removing the need for manual formula design and enabling targeted optimization toward context-dependent unsafe behaviors. In addition, the proposed framework could be extended to evaluate autonomous driving policies under adversarial or rare-event conditions by optimizing latent variables adversarially with respect to the control policy of interest.

ACKNOWLEDGMENTS

This work has been partially supported by the iNEST project funded by the European Union Next-GenerationEU (PNRR – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043) and by the National Science Foundation through the following grants: CAREER award (SHF-2048094), CNS-2039087, IIS-SLES-2417075, and funding by Toyota R&D through the USC Center for Autonomy and AI. This work does not reflect the views or positions of any organization listed.

REFERENCES

[1] Ezio Bartocci, Luca Bortolussi, Michele Loreti, and Laura Nenzi. 2017. Monitoring mobile and spatially distributed cyber-physical systems. In *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design*. ACM, Vienna Austria, 146–155. https://doi.org/10.1145/3127041.3127050

[2] Francesca Cairoli, Luca Bortolussi, Jyotirmoy V Deshmukh, Lars Lindemann, and Nicola Paoletti. 2025. Conformal Predictive Monitoring for Multi-modal Scenarios. In *International Conference on Runtime Verification*. Springer, 336–356.

[3] Francesca Cairoli, Nicola Paoletti, and Luca Bortolussi. 2023. Conformal quantitative predictive monitoring of stl requirements for stochastic processes. In *Proceedings of the 26th ACM international conference on hybrid systems: computation and control*. 1–11.

[4] Luigi Di Lillo, Tilia Gode, Xilin Zhou, Margherita Atzei, Ruoshu Chen, and Trent Victor. 2024. Comparative safety performance of autonomous-and human drivers: A real-world case study of the Waymo Driver. *Heliyon* 10, 14 (2024).

[5] Francesco Giacomarra, Mehran Hosseini, Nicola Paoletti, and Francesca Cairoli. 2025. Certified Guidance for Planning with Deep Generative Models. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 877–885.

[6] Luis Gressenbuch and Matthias Althoff. 2021. Predictive monitoring of traffic rules. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 915–922.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 6840–6851.

[8] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Chen Lv, and Jaime Fernández Fisac. 2024. Versatile behavior diffusion for generalized traffic agent simulation. *arXiv preprint arXiv:2404.02524* (2024).

[9] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. 2024. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1334–1345.

[10] Lars Lindemann, Xin Qin, Jyotirmoy V Deshmukh, and George J Pappas. 2023. Conformal prediction for stl runtime verification. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*. 142–153.

[11] Henry X Liu and Shuo Feng. 2024. Curse of rarity for autonomous vehicles. *nature communications* 15, 1 (2024), 4808.

[12] Oded Maler and Dejan Nickovic. 2004. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 152–166.

[13] Laura Nenzi, Ezio Bartocci, Luca Bortolussi, and Michele Loreti. 2022. A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems. *Logical Methods in Computer Science* 18 (2022).

[14] Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nick Roy. 2023. Scenario diffusion: Controllable driving scenario generation with diffusion. *Advances in Neural Information Processing Systems* 36 (2023), 68873–68894.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[16] Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. 2025. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 17207–17218.

[17] Davide Scassola, Sebastiano Saccani, Ginevra Carbone, and Luca Bortolussi. 2025. Zero-shot conditioning of score-based diffusion models by neuro-symbolic constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 20302–20309.

[18] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. pmlr, 2256–2265.

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

[20] Qunying Song, Kaige Tan, Per Runeson, and Stefan Persson. 2023. Critical scenario identification for realistic testing of autonomous driving systems. *Software Quality Journal* 31, 2 (2023), 441–469.

[21] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

[22] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. 2023. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7280–7290.

[23] Yixiao Wang, Chen Tang, Lingfeng Sun, Simone Rossi, Yichen Xie, Chensheng Peng, Thomas Hannagan, Stefano Sabatini, Nicola Poerio, Masayoshi Tomizuka, et al. 2024. Optimizing diffusion models for joint trajectory prediction and controllable generation. In *European Conference on Computer Vision*. Springer, 324–341.

[24] Nick Webb, Dan Smith, Christopher Ludwick, Trent Victor, Qi Hommes, Francesca Favaro, George Ivanov, and Tom Daniel. 2020. Waymo's safety methodologies and safety readiness determinations. *arXiv preprint arXiv:2011.00054* (2020).

[25] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. 2023. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493* (2023).

[26] Chejian Xu, Aleksandr Petiushko, Ding Zhao, and Bo Li. 2025. DiffScene: Diffusion-Based Safety-Critical Scenario Generation for Autonomous Vehicles. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 8 (2025), 8797–8805.

[27] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. 2023. Guided Conditional Diffusion for Controllable Traffic Simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3560–3566.

[28] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. 2023. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 17863–17873.