CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Target-Guided Trajectory Generation for Controllable Traffic Scenarios

A novel conditioning method for diffusion-based trajectory generation enabling controllable traffic scenario synthesis

Master's thesis in Computer science and engineering

Jacob Bredin

Linus Haraldsson

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Target-Guided Trajectory Generation for Controllable Traffic Scenarios

A novel conditioning method for diffusion-based trajectory generation enabling controllable traffic scenario synthesis

Jacob Bredin
Linus Haraldsson

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Target-Guided Trajectory Generation For Controllable Traffic Scenarios
A novel conditioning method for diffusion-based trajectory generation enabling controllable traffic scenario synthesis

Jacob Bredin
Linus Haraldsson

Supervisor: Adam Breitholtz, Computer Science and Engineering

Advisors: Zhennan Fei, Volvo Cars
Sadegh Rahrovani, Volvo Cars
Andreas Tingberg, Volvo Cars

Examiner: Fredrik Johansson, Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Generated trajectories of agents in a intersection with guided vehicle in red.

Typeset in LaTeX
Gothenburg, Sweden 2025

Target-Guided Trajectory Generation For Controllable Traffic Scenarios
A novel conditioning method for diffusion-based trajectory generation enabling controllable traffic scenario synthesis

Jacob Bredin
Linus Haraldsson

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Autonomous vehicles (AVs) must be evaluated under rare and hazardous driving conditions to ensure safety and reliability. However, creating such safety-critical scenarios is made difficult for several reasons. They occur infrequently in real-world data and are costly to reproduce through physical testing, while existing simulation methods often yield unrealistic behaviors. This thesis explores generative modeling as a tool for producing realistic and controllable scenarios for closed-loop evaluation of AV systems.

We introduce a novel diffusion-based method for generating adversarial trajectories, with a focus on Classifier-Free Guidance (CFG) to steer agents toward defined targets. The approach incorporates target information during training, uses data augmentation to improve robustness, and applies trajectory optimization to enhance accuracy. Building on the Versatile Behavior Diffusion (VBD) framework, our method strengthens controllability while preserving realistic motion patterns.

The experimental results show that CFG improves guidance performance without any additional computational cost during inference, which has been a major limitation of prior approaches, while still matching the accuracy of classifier-based guidance. When combined with classifier-based guidance, CFG yields substantial improvements in target accuracy and reduces the number of required guidance iterations. Furthermore, direct trajectory optimization is shown to further refine target accuracy, although it introduces trade-offs with respect to adherence to traffic regulations. Collectively, these findings establish an efficient and versatile framework for the generation of safety-critical driving scenarios, thereby advancing the methodological foundation for rigorous evaluation of autonomous vehicle systems.

Keywords: Scenario Generation, Guided Trajectory Generation, Diffusion, Deep learning, Machine learning.

Acknowledgements

We would like to extend our gratitude to our supervisors at Volvo Cars and at Chalmers University of Technology. You have provided us with a valuable source of support and guidance throughout our thesis work. Giving us the opportunity to discuss and untangle problems together, providing both insights and subject specific knowledge. Thank you.

Jacob Bredin and Linus Haraldsson, Gothenburg, 2025-12-10

Contents

List of Figures

List of Tables

1

Introduction

Scenario-based testing has become an integral part of the evaluation of advanced driver-assistance systems and autonomous driving software in modern vehicles. The software control systems are validated within interactive simulation environments, where a diverse set of traffic scenarios can be constructed and executed. Scenarios are designed to evaluate general and individual features of the control software under different conditions.

For example, a test-scenario used for evaluating the automatic braking system can consist of a road crossing, where a pedestrian or another vehicle suddenly moves in front of the vehicle being tested. If the tested vehicle manages to brake without colliding, the automatic braking system completes the test successfully.

Scenarios used in testing are primarily created by human experts, where agents follow deterministic rules that control their behavior. This results in high controllability and reproducibility, but may lack the diversity and interactivity of real traffic, since all behaviors needs to be defined manually.

Evaluating the control software on a wide range of scenarios, both safe and unsafe, is necessary for obtaining a satisfactory test coverage. Safety-critical scenarios is a category that is of particular interest, since they allow repeated testing of difficult and dangerous situations, that is both cost effective and safe, that can substantially increase vehicle safety.

Creating a catalog of safety-critical scenarios is made difficult by their rare and diverse nature. Collecting them from traffic would require an infeasible amount of real-world data, that would then need to be processed to filter out interesting scenarios. Beyond collection, ensuring sufficient coverage poses an additional challenge: the dataset must contain a wide range of diverse and relevant scenarios that enable comprehensive testing of autonomous vehicle capabilities.

Datasets capturing the movements of vehicles, pedestrians, and other agents in real-world traffic are used to train machine learning models. These models are then able to generate artificial yet realistic and interactive traffic scenarios. As recorded data primarily consist of regular, safe traffic, the models learn to produce similar behaviors. These models are therefore once again insufficient for generation of safety-critical scenarios. However, These models can serve as realism priors, informing any safety critical scenario generation method how a scenario would normally play out if everything behaved in a non-critical way.

Several papers have investigated how safety-critical scenarios can be created. These methods primarily focus on adversarial methods, either using optimization methods or reinforcement learning to create targeted attacks from one vehicle to another. While this results in a high collision rates, it has no guarantees of resembling real-world safety-critical events. Another issue is the diversity of this type of attack, it only captures a small number of behaviors that are significantly biased directly towards creating collisions, often trained adversarially against a single model, or unguided cars. Therefore, further research must be conducted to enable a wider variety of scenarios of differing criticality to be created.

This work addresses these challenges by focusing on controllable yet realistic scenario generation. The proposed model is capable of guiding vehicle behavior towards specified spatial, temporal, and velocity constraints, while preserving the flexibility to generate trajectories from arbitrary initial conditions. This design accommodates both manual and automated scenario creation. As the trained model learns to guide agents towards targets, most behavioral bias depends on the guidance specification process, whether defined by a user or an algorithm. As a result, the model can generate diverse behaviors without retraining. In turn, enabling the design of safety-critical scenarios for targeted testing of specific autonomous vehicle behaviors, thereby supporting the development of more efficient testing frameworks.

1.1 Problem Specification

The validation of autonomous driving systems requires access to realistic and diverse traffic scenarios. Generative models based on real-world data provide a strong realism prior, but they typically reproduce conventional safe behaviors and offer limited control over the outcome of generated scenarios. Adversarial approaches can enforce safety-critical events, but they are often biased, narrow in scope, or computationally expensive. What is lacking is a controllable and computationally efficient generative framework that balances realism with the ability to target specific scenario outcomes.

This thesis investigates the problem of **controllable traffic scenario generation**, with a focus on improving target accuracy and computational efficiency in diffusion-based motion planning models. The central aim is to enable reliable guidance of agent trajectories toward specified spatial, temporal, and velocity constraints while maintaining the naturalism of interactive traffic flow. The research is guided by the following questions:

1. How can diffusion-based motion planning models be guided to achieve precise realization of user-specified trajectory constraints?
2. What methods can be employed to improve the computational efficiency of such guidance, enabling their use in large-scale and closed-loop testing settings?
3. To what extent is the performance of unguided traffic generation degraded?

The specific contribution of this thesis is stated in the list below.

Summary of contributions:

- A new method for guiding diffusion-based motion planning models using classifier-free guidance.
- Significantly improved guidance performance compared to the baseline Versatile Behavior Diffusion model [1].
- A comparison of the computational effectiveness of different guidance methods, showing an up to 20x runtime improvement.

2

Theory

This chapter goes into detail on different aspects of scenario generation and preliminary information for the methods used later in this thesis.

2.1 Scenario Representations

For motion planning, it is common to use a simplified representation of the scene, reducing the amount of detail to road markings, traffic lights and agents. In this thesis we will be working with Waymo's Open Motion Dataset (WOMD) as our source of training and validation data. Waymo's dataset is represented using two main components: dynamic agents (e.g., vehicles, cyclists, and pedestrians) and static map elements (e.g., lanes, crosswalks, and road boundaries). Dynamic agents are represented as oriented bounding boxes parameterized by their state and geometry. Each agent is represented as $\{x, y, yaw, v_x, v_y, l, w, h\}$ at each timestep. Table 2.1.1 gives an explanation for each element in the agent state. In total there are 91 states for each agent, in a scenario, with each scenario spanning 9.1 seconds sampled at 10 Hz. This is split into a one second history, a starting timestep and an eight second future. The dataset includes up to 128 agents per scene.

Element	**Explanation**
x, y	Center coordinates of the agent in the global frame.
yaw	Heading angle of the agent, measured counterclockwise from the x-axis.
v_x, v_y	Velocity components of the agent along x and y axes.
l, w, h	Length, width and height of the bounding box enclosing the agent.

Table 2.1.1: Representation of dynamic agents as oriented bounding boxes.

The static environment contains the geometry of the road network. It is represented as a collection of polylines, where each polyline corresponds to a road element such as a lane centerline, lane boundary, or crosswalk outline. A polyline is defined by a sequence of connected points, with each point parameterized as $\{x, y, u_x, u_y, type, id, valid\}$. Table 2.1.2 explains the meaning of each parameter.

Element	Explanation
x, y	Position of the point in the global coordinate system.
u_x, u_y	Unit vector indicating direction.
$type$	Semantic type of the element (e.g., lane centerline, crosswalk, lane boundary).
id	Unique identifier of the polyline to which the point belongs.
$valid$	Boolean flag indicating whether the point is active.

Table 2.1.2: Representation of static map elements as polylines.

There are multiple choices of coordinate systems when working with motion data. By default in WOMD, the data is represented in a global frame of reference, where the road graph is stationary and agents change position for each timestep. Another common choice is to use a local frame of reference, positioned with the ego-vehicle at the center, either for the first timestep or for every timestep. More generally, a local frame of reference can be used for any agent or element in the scene. The local reference frame is obtained by translating and rotating all elements with respect to a specific agent or scene element. A less common choice is a relative frame of reference, where the position, yaw and velocity from a reference agent is subtracted from all objects in the scene, where applicable, without rotation.

2.2 Motion Prediction and Conditioning

Generating plausible future trajectories for agents in a scenario can be described as sampling from a probability distribution: $\Pr(\boldsymbol{x}|\boldsymbol{c})$. The probability distribution describe all possible future trajectories $\boldsymbol{x}$ and is conditioned on the scene context $\boldsymbol{c}$. The scene context contain information about agents' history and scene elements, such as road markings and traffic lights. Using this mathematical description, we will expand on three interesting methods for how generating trajectories can be described.

The first method is to sample joint trajectories [2], where agents behave like regular traffic. They avoid collisions, follow markings and traffic rules. This is useful for testing autonomous driving algorithms in safe and common traffic situations. The second method generate marginal trajectories [3]. These are plausible trajectories of every agent individually, meaning the trajectories do not account for interactions with other agents. This has been used in the literature to sample probable, unsafe trajectories that can be used to create safety-critical scenarios [1]. The third method extends what the trajectories are conditioned on, allowing for further control of what agents do, or specific outcomes. For example, an agent can be conditioned to travel through a specific position or a whole predetermined trajectory. This method has both been used for generating adversarial scenarios [1] and for introducing more control over scenario outcome [4].

2.3 Open and Closed-Loop Simulation

To allow an autonomous driving (AD) algorithm to be tested in an interactive environment, there needs to be feedback between it and the generative model, this is known as a closed-loop simulation. This is done by stepping the scenario forward in time and passing the new initial conditions to the generative model. New trajectories are predicted and the next time step is passed back to the AD algorithm. Due to the time-cost of generating new trajectories, it is common to update them with a one second gap or more. The environment becomes less responsive as the gap increases. If the trajectories are only planned at the start, it is called open-loop simulation. Selecting how often the trajectories are updated is a trade-off between being able to test more scenarios and scenario quality. A common issue with closed-loop generation is the cumulative error that grows as the simulation proceeds. Resulting in vehicles not following road markings etc. This can limit the length of a test-scenario, and hence the usefulness of the test. Therefore it is important to reduce the cumulative error when testing AD algorithms.

2.4 Diffusion Models

Diffusion models have emerged as a powerful and widely adopted generative modeling technique in recent years, fundamentally transforming the landscape of image synthesis [5]. Unlike earlier generative approaches such as Generative Adversarial Networks (GANs) [6], diffusion models synthesize images by learning to reverse a gradual noise corruption process inspired by thermodynamic diffusion. Through this progressive denoising procedure, they achieve stable training dynamics and high-quality image generation. These characteristics have established diffusion models as a leading framework for image generation across a wide range of domains.

Beyond producing realistic images, diffusion models offer remarkable flexibility and controllability through various conditioning mechanisms. This capability has led to their adoption in specialized applications such as controllable traffic scene generation and autonomous driving simulation [1, 7, 8, 9]. In these contexts, diffusion-based methods enable the synthesis of realistic and diverse traffic scenarios with control over scene attributes, making them a valuable tool for research and development in safety-critical and data-driven transportation systems.

To formalize the generative process, diffusion models are typically described through a forward diffusion process and a corresponding reverse denoising process. The forward process gradually adds Gaussian noise to the data in a series of steps, effectively destroying structure and transforming the original data distribution into a standard normal distribution. It is defined recursively as:

$$q(x_t|x_{t-1}) = \sqrt{\beta_t}x_{t-1} + (1-\beta_t)\epsilon, \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{I})$$

The parameter β_t is known as the noise-schedule and is designed to gradually move the original data distribution to the standard normal distribution in T diffusion steps. Since each diffusion step only depends on the previous step, it is a Markov

chain. This means that the joint distribution for the complete diffusion process can be rewritten as a product of probabilities:

$$q(x_0, x_1, ..., x_T) = q(x_0) \prod_{t=1}^{T} q(x_t|x_{t-1})$$

Since ϵ follows a Gaussian distribution, the forward process can be expressed in closed form. To simplify the notation we let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$.

$$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1)$$

We would like the reader to notice the structure of the closed form function and that $\bar{\alpha}_t$ is chosen to range from one to zero. This means the original data is interpolated towards a standard normal distribution.

The Reverse process is also described by a Markov chain where the mean and standard deviation is parametrized by a neural network.

$$p_\theta(x_{t-1}|x_t) = \mu_\theta(x_t, t) + \sigma_\theta(x_t, t)$$

A common choice in the literature is to parameterize this model to have a fixed schedule for the variance term: $\sigma_\theta(x_t, t) = (1 - \beta_t)\epsilon$, $\epsilon \sim \mathcal{N}(0, 1)$. By parameterization μ_θ as $\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$, where $\epsilon_\theta(x_t, t)$ is the predicted noise given parameters θ for noisy sample x_t at timestep t, the learning objective is therefore to recover the noise added during the forward pass.

The sampling process is then to remove noise for T time-steps. With this choice of the parameterization the loss becomes:

$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta\left(x_t,\, t\right)\|^2\right]$$

2.4.1 Classifier Guidance

A benefit of using diffusion models is that they can easily be conditioned to generate specific output without retraining the model. Originally introduced by P. Dhariwal and A. Nichol [10] classifier guidance applies a classifier $p_\phi(y|x_t, t)$ to the output of an image diffusion model to obtain a gradient $\nabla \log\ p_\phi(y|x_t, t)$.

The gradient is used to guide the diffusion process, by sampling:

$$p_{\theta,\phi}(x_t \mid x_{t+1}, y) \propto p_\theta(x_t \mid x_{t+1})\, \nabla \log\ p_\phi(y|x_t, t)$$

throughout the denoising process, the learned prior of the diffusion model is steered towards the desired output via the classifier.

The method is universal and any differentiable function $f(\cdot)$ could in theory be used to guide a diffusion model, as long as the target is within the learned distribution of the diffusion model. In practice, the function $f(\cdot)$ also needs to be able to score noisy outputs in a meaningful way during the reverse process. In the context of motion planning, guidance functions are often defined to minimize the distance to a target or agent to e.g. guide, or discourage collisions during trajectory generation [1, 7, 9]. The level of noise in generated actions does not hinder the calculation of distances from the resulting trajectory.

2.4.2 Classifier-Free Guidance

As an alternative to classifier guidance, J. Ho and T. Salimans [5] introduced a method called *classifier-free guidance*. In this approach, the explicit classifier is removed, and the diffusion model itself is trained to handle both conditioned and unconditioned generation. The conditional information typically comes from the training label, which could be a class-label when training on a dataset with pictures of animals. Alternatively, it could be used for preserving information in a generated image, such as to instruct the diffusion model to generate a specific object on part of the image.

Formally, the denoising model is defined as $\epsilon_\theta(x_t, t, c)$, where c denotes an optional conditioning variable (e.g., a class label, text embedding, or trajectory goal). During training, c is randomly dropped with a fixed probability, allowing the model to learn both conditional $p_\theta(x_t \mid x_{t+1}, c)$ and unconditional $p_\theta(x_t \mid x_{t+1})$ behaviors.

At inference time, classifier-free guidance combines these two predictions using a weighting factor $w \geq 1$, producing the guided prediction $\tilde{\epsilon}_\theta(x_t, t, c)$, where $\epsilon_\theta(x_t, t)$ is the model output without conditioning:

$$\tilde{\epsilon}_\theta(x_t, t, c) = \epsilon_\theta(x_t, t, \emptyset) + w\Big(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t)\Big)$$

The resulting distribution can be interpreted as:

$$p_\theta(x_t \mid x_{t+1}, c; w) \propto p_\theta(x_t \mid x_{t+1})^{1-w}\, p_\theta(x_t \mid x_{t+1}, c)^w.$$

Increasing w amplifies the influence of the conditioning variable, leading to more specific but potentially less diverse samples. Conversely, smaller w values yield more diverse but less targeted outputs.

This approach provides a simple yet powerful mechanism for controlling specificity and diversity without requiring an external classifier. Later in this thesis, we demonstrate that classifier-free guidance translates effectively to motion planning and enhances controllability in trajectory generation.

2.5 Motion Planning Metrics

To measure the quality of generated scenarios, a range of different metrics have been used in the literature, often serving as proxy metrics for realism. A popular choice are the metrics from the Waymo Open Sim Agents Challenge (WOSAC) [1, 11]. The metrics are known as: average displacement error (ADE), collision, off-road, and wrong-way.

The collision, off-road and wrong-way metrics calculates the proportion of their respective events aggregated over all agents and time steps within a scenario. As these are boolean events, the values ranges between 0 and 1. While ADE is similarly computed over all steps and agents, there is no max value as it is a distance metric.

2.5.1 Average Displacement Error

Average displacement error is the average distance between each point of the ground truth trajectory and the generated trajectory. The variable A denotes the number of agents and T is the total number of future timesteps. The points of the ground truth trajectory is denoted by $x_{i,t}$ and the generated trajectory points are denoted by $\hat{x}_{t,i}$.

$$\text{ADE} = \frac{1}{AT} \sum_{i=0}^{A} \sum_{t=0}^{T} \left\| x_{i,t} - \hat{x}_{i,t} \right\|_2$$

2.5.2 Collisions

The collision metric computes the proportion of agents with one or more instances of overlapping bound boxes. The notation bb_i represents the bounding box of agent i. The overlap function computes whether two different bounding boxes overlap.

$$\text{overlap}(bb_1, bb_2) = \begin{cases} 1, & \text{if the bounding boxes are overlapping.} \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{Collisions} = \frac{1}{A} \sum_{i=0}^{A} \bigvee_{\substack{j=0, \\ j \neq i}}^{A} \bigvee_{t=0}^{T} \text{overlap}(bb_{i,t}, bb_{j,t})$$

2.5.3 Off Road

The off Road metric is derived using the offroad function. This function determines whether a vehicle-type agent is on the road by evaluating the signed distance from each bounding box corner to the nearest point on the road graph R. For each bounding box corner of the agent. If all bounding box corners have a positive signed distance, the vehicle is outside of the road and the function returns 1, otherwise 0.

$$\text{offroad}(bb, R) = \begin{cases} 1, & \text{if all corners of the bounding box are located outside the nearest road.} \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{Off Road} = \frac{1}{A} \sum_{i=0}^{A} \bigvee_{t=0}^{T} \text{offroad}(bb_{i,t}, R)$$

Wrong Way

The measurement of whether vehicle-type agents are driving on the correct side of the road, or driving into oncoming traffic.

$$\text{wrongway}(bb, R) = \begin{cases} 1, & \text{if the bounding box is oriented in the opposite travel direction of the nearest road.} \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{Wrong Way} = \frac{1}{A}\sum_{i=0}^{A}\bigvee_{t=0}^{T}\text{wrongway}(bb_{i,t}, R)$$

2.6 Versatile Behavior Diffusion

Introduced in 2024 by Z. Huang et al [1], the motion planning model Versatile Behavior Diffusion (VBD) achieved state-of-the-art performance on Waymo's 2024 motion prediction challenge [12] and Waymo's 2024 sim agents challenge [13]. The model is capable of generating joint motion planning for agents that adhere to scene context, resulting in realistic and interactive scenarios. It can also generate marginal trajectories to predetermined anchor points for every agent.

The authors demonstrated that classifier guidance can be used for controlling the behavior of agents during inference, without requiring additional fine-tuning. By defining differentiable reward functions, such as minimizing an agents distance to a target, the model can be guided to generate trajectories that maximize these rewards. The model can also be used for closed-loop generation, allowing agents to react and avoid collisions from an externally controlled agent. Combining these methods, the authors highlight the potential use of the model for generating safety-critical scenarios for testing autonomous driving algorithms.

2.6.1 Architecture Overview

VBD has an encoder-decoder architecture that is comprised of one scene encoder and two motion planning decoders, depicted in figure 2.6.1. The double decoder architecture is motivated by an observed improvement in training loss stability and improvements in scenario realism during inference [1]. The primary decoder is known as the Denoiser and uses 50 diffusion steps to generate joint future actions for all agents at once. The secondary decoder is called the Behavior Predictor and uses a single forward pass to predict the marginal future actions of all agents. The marginal actions are computed with respect to 64 anchor points that are used as end points. The anchor points are computed from the training dataset by clustering all end points using K-means clustering, for each agent type separately. Both decoder types use a kinematic bicycle model to turn the generated actions into trajectories.

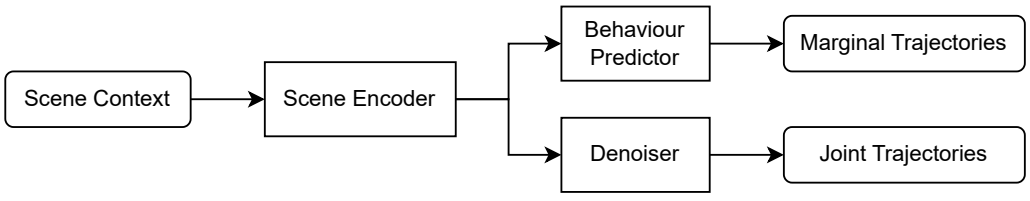


Figure 2.6.1: The VBD encoder-decoder structure.

2.6.2 Preprocessing

To generate trajectories, VBD uses a small subset of the original scenario data. The scenario is preprocessed to reduce and transform the data for the agents, the roadgraph and the traffic lights. The output from the preprocessing step is the scene context and the relational tensor.

Preprocessing starts by selecting a maximum of 32 agents (A) from the scene based on their proximity to the ego agent. To represent the agents, information about their respective type, velocity and bounding box dimensions are extracted. Next, the roadgraph polylines are sorted by their distance to each agent. The 256 nearest polylines (P) are kept, while the rest are discarded. This method of downsampling is not perfect and important map features can be removed in place of less important ones. Polylines are represented by a varying number of points and the chosen polylines are downsampled further by selecting 30 points with uniform spacing along each of them. This complete downsampling procedure reduces the number of points in a scenario from 30 000 to a maximum of 7 680 points. The points are additionally transformed to lie in the local reference frame of the first point in each polyline. In the original dataset, traffic lights are connected to polylines via an ID. To associate the traffic light state directly to each polyline, the traffic light state is joined with the downsampled polyline data and the polyline's type. For the traffic lights, a maximum of 16 lights (L) are kept based on their proximity to the selected agents. Information about their global position and traffic light state are extracted. In table 2.6.1 a summary of the scene elements, the number of element type and their representation are shown.

Scene Element	**Variable**	**Number**	**Information**
Agent	32	A	$\{agent\ type, v_x, v_y, l, w, h\}$
Polyline	256	P	$\{type, local\ points, traffic\ light\ state\}$
Traffic Light	16	L	$\{x, y, traffic\ light\ state\}$

Table 2.6.1: Scene context element representations.

The second stage of preprocessing computes the relations tensor. It contains the scene transformed into every local reference frame of all scene element. This gives a symmetric representation of the poses in the scene without any bias towards any particular scene element. The relations tensor is represented by a real valued tensor of shape: $\mathbb{R}^{(A+P+L)\times(A+P+L)\times(x,y,yaw)}$ or $\mathbb{R}^{304\times304\times3}$ with the numbers given above. Agent's use the current timestep pose. The polylines are represented by their first points pose. The traffic lights use their position and a yaw of 0. This means every traffic light points towards the east in the global coordinate system. This is a compromise made since traffic lights have no yaw in the original dataset.

Scene Element	Number	Relational Information
Agent	32	$\{(x_{i,0}, y_{i,0}, yaw_{i,0}), (x_{i,1}, y_{i,1}, yaw_{i,1})\ ...\ \}, i = 0, ..., 31$
Polyline	256	$\{(x_{j,0}, y_{j,0}, yaw_{j,0}), (x_{j,1}, y_{j,1}, yaw_{j,1})\ ...\ \}, j = 32, ..., 287$
Traffic Light	16	$\{(x_{k,0}, y_{k,0}, yaw_{k,0}), (x_{k,1}, y_{k,1}, yaw_{k,1})\ ...\ \}, k = 288, ..., 303$

Table 2.6.2: The relations tensor.

2.6.3 Scene Context Encoder

Figure 2.6.2 shows a diagram of the scene encoder. It is comprised of four parts: the scene element input and relational input in the first column, the scene context encoder in the second column, concatenation of the intermediate encodings in the third column and the transformer encoder that produce the final embeddings in the fourth.

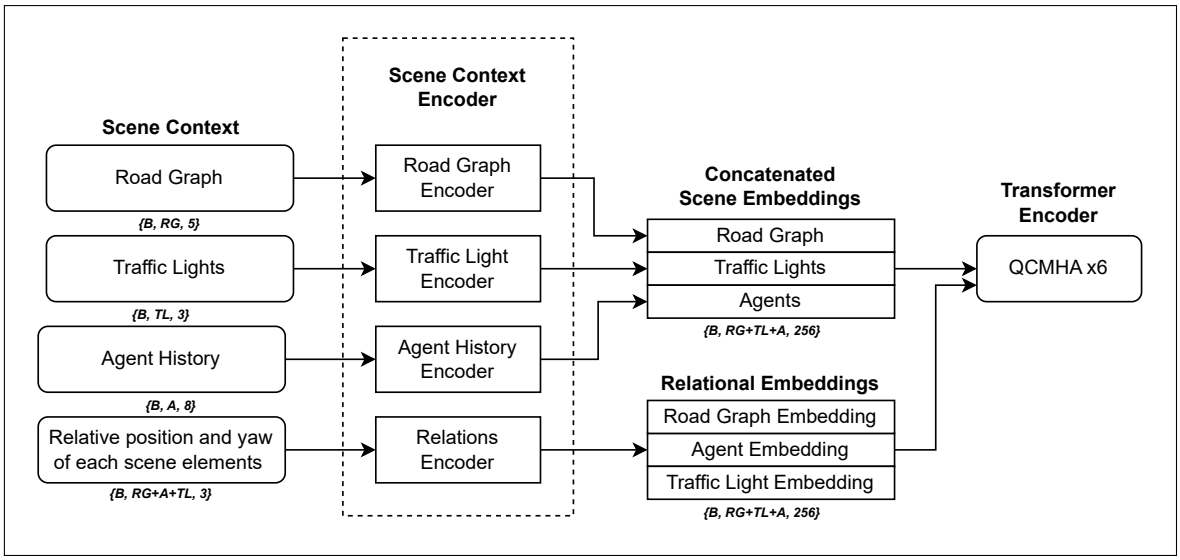


Figure 2.6.2: VBD Scene Encoder.

The scene context encoder embed the preprocessed scene elements using separate MLPs for the agents, the roadgraph and the traffic lights. The relational tensor is encoded differently using a Fourier encoder. This is to enable the network to model high-frequency variations in low dimensional domains (x, y, yaw). Each input dimension is expanded using a Fourier feature mapping followed by a per-dimension MLP. This enhances representational capacity compared to direct raw inputs [14].

$$\mathbf{X} = 2\pi\, \mathbf{x}_{\text{in}} \mathbf{W}_f$$

$$\mathbf{X}_{\text{emb}} = [\cos(\mathbf{X}), \sin(\mathbf{X}), \mathbf{x}_{\text{in}}]$$

$$\mathbf{y} = \sum_{i=1}^{D} f_i(\mathbf{X}_{\text{emb}}^{(i)})$$

Here, $\mathbf{W}_f$ denotes learnable frequency weights, and f_i represents the i-th per-dimension MLP. The outputs are summed to obtain the final representation.

The scene element and relational embeddings are separately concatenated into tensors of shape $\mathbb{R}^{(A+P+L)\times 256}$, that are passed onto the transformer encoder. The transformer encoder uses query-centric multihead attention (QCMHA) to compute rich context embeddings for all scene elements using the complete scene context. Figure 2.6.3 shows a diagram of the architecture for a single attention head in QCMHA. The architecture is an extension of scaled dot-product attention, with the extension highlighted in blue in the diagram. The modification allows self-attention to be computed for the scene context together with the relations tensor, that contain pose information about each token. The diagram shows query-centric self-attention, but it can also be extended to cross-attention.

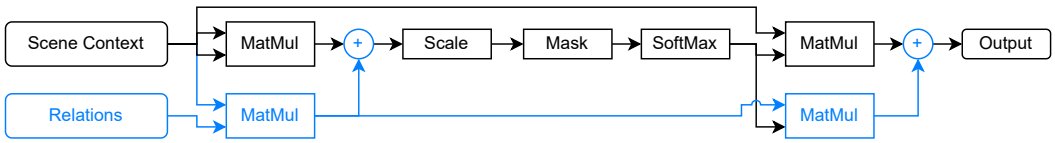


Figure 2.6.3: Query-Centric Attention with difference to scaled dot-product attention highlighted in blue.

2.6.4 Denoiser

The denoiser uses a transformer architecture with a combination of modified self-attention and modified cross-attention layers to denoise agent actions. Figure 2.6.4 depicts the full denoiser architecture. The denoiser processes the noisy actions for 50 steps to produce the final, clean actions. At the beginning of each pass, an embedding of the current diffusion noise level is added to all action sequence tokens. A timestep embedding is also added. In each attention block: block 1 and 2, every agent action sequence, denoted by A_i are processed separately. Self-attention is computed between one agent's action sequence and all other agent's action sequences A. The relations tensor associated with the current agent, denoted R_i, is used in combination with the other agent's actions sequences, as shown along the bottom of the figure. A causal mask is applied to only allow the current agent's tokens to attend to themselves and past action sequence tokens of other agents through time. In the cross-attention layer, the current agent's action sequence is processed together with all scene elements, denoted S, and the current agent's full relations tensor R_i. The figures shows the values for the query, key and value.

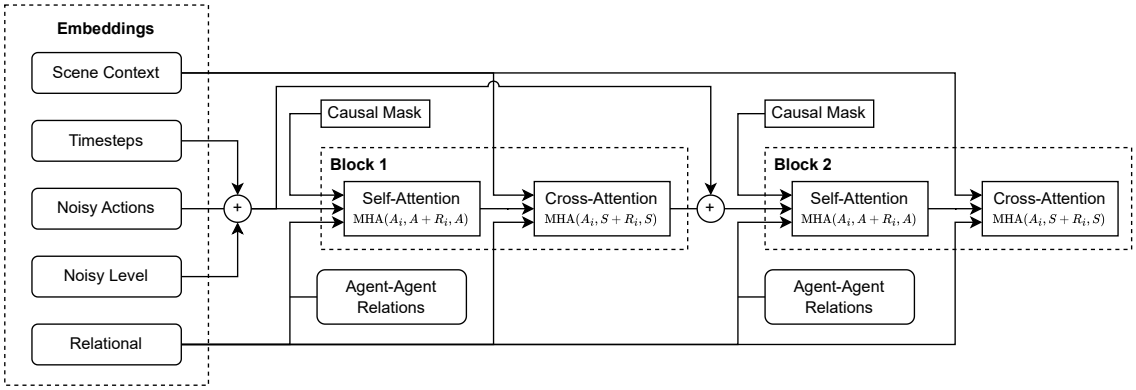


Figure 2.6.4: VBD Denoiser.

3

Related Work

The challenge of generating realistic and safety-critical traffic scenarios is complex, with a multitude of different solutions and variations of possible trajectories to generate. This section will summarize and categorize these solutions to show how different works in the related literature have approached the problem, and to highlight their respective strengths and limitations.

3.1 Datasets

To be able to generate realistic traffic scenarios using deep generative methods, high quality datasets are required to learn a diverse and realistic representation. Within the related literature, there are two open datasets that stand out.

The largest dataset, and recorded over the most hours driven, is the Waymo Open Dataset (WOMD) [15, 16]. This is a dataset collected from real-world traffic scenarios represented as a map of the road with bounding-boxes and trajectories for all agents on the road. This dataset has been widely used in the Waymo Open Sim Agents Challenge [11] (WOSAC). This challenge has been a driving force behind realistic autoregressive trajectory generation, as it requires agents to replan every 10 steps. Many recent publications are focused on this task. [1, 2, 3]

Another commonly used dataset is the NuScenes dataset [17], which represents scenarios in a similar way to WOMD, with a distinction in how the map is represented. This dataset has also been widely used, and outside of the WOSAC competition might be the most common [7, 9, 18]. Other available dataset such as the dataset from Lyft [19] are also available, but are less commonly used in the scenario generation literature.

The main difference between the WOMD and NuScenes is how they represent the map. The NuScenes dataset represents an area as an birds-eye-view image. The number of channels representing different semantic categories, e.g. road, sidewalk, crosswalk, lanes and more. In WOMD, the map is instead represented as road edges and other markings in the form of polylines, creating a denser representation of the data, only containing relevant information.

3.2 Realistic Scenario generation

Recent generative approaches leverage imitation learning to model human driving behaviors and generate realistic traffic scenarios. Prominent methods in this domain include diffusion models [1, 9, 18], auto-regressive generative transformers, [2], and multi-modal trajectory-generating transformers [3].

3.2.1 Transformer-based Motion Prediction

The Motion Transformer (MTR) framework [20] for motion prediction in autonomous driving employs a transformer-based encoder-decoder architecture with learnable intention queries to accurately predict future trajectories for a single agent. It combines global intention localization with local movement refinement. Building on this, MTR++ extends the framework to handle multi-modal, multi-agent prediction through symmetric scene context modeling and mutually guided intention querying [3].

In particular, symmetric scene context modeling improves interaction reasoning by processing the environment symmetrically from each agent's perspective. Rather than relying on a global coordinate system, each polyline (representing agent trajectories and road segments) is mapped into a local coordinate frame via a learnable transformation. This design decouples input features from a fixed frame, allowing for symmetry, preventing biases inherent in 'ground truth' environment encodings, which are often centered around the ego vehicle. To relate these polylines, MTR++ employs a query-centric self-attention mechanism: for each query token, all other tokens are transformed into the query's local coordinate system, allowing the model to compute relational features such as relative positions and headings. These features are embedded into the attention process, enabling efficient interaction modeling across agents.

3.2.2 Controllable Diffusion methods

While MTR++ emphasizes multi-agent prediction accuracy, other works focus on controllability in scenario generation. Conditional Trajectory Generation (CTG) [18] introduces a diffusion-based model for traffic generation, that can, during inference, be guided using a classifier-guidance-like setup. Instead, CTG replaces the classifier with differentiable Signal Temporal Logic (STL) constraints. These constraints encode temporal and logical rules directly into the guidance mechanism, enabling trajectories to be generated that more reliably satisfy user-specified behavioral requirements. This approach increases scenario diversity by supporting the controlled generation of edge cases and out-of-distribution behaviors.

Versatile Behaviour Diffusion (VBD) [1], extends the classifier-guidance reformulation introduced by CTG. Instead of defining constraints using STL, VBD employs reward and penalty functions to guide behavior, allowing for more flexible specification of desired outcomes. This method enhances controllability while supporting diverse and coherent scenarios. Moreover, VBD builds upon the architectural ad-

vances of MTR++, incorporating its symmetric scene context modeling and mutually guided intention querying to capture complex multi-agent interactions. This yields a framework that not only offers fine-grained controllability through diffusion-based guidance, but also achieves state-of-the-art scenario realism, as reflected in its second-place ranking at the WOSAC 2024 challenge [11].

3.2.3 Computational Inefficiency of Guidance Methods

Both CTG's and VBD's guidance methods involve iterative optimization within the diffusion loop, requiring a model call and backpropagation through the model. Algorithm 1 illustrates the nested loops and computationally relevant steps when utilizing VBD's guidance paradigm, leading to slow inference.

To address this limitation, SceneDiffuser [4] introduces a novel guidance formulation by treating trajectory generation as an inpainting task. Constraints are encoded as masked regions which the model learns to complete, enabling the integration of both hard and soft behavioral constraints. In addition to this, SceneDiffuser also employs amortized diffusion, distributing the denoising process across trajectory time steps for improved efficiency, with a small sacrifice in generation quality.

Note: Model inference and gradient computation are comparatively heavy.

Require: t: timesteps, D: diffusion steps, G: guidance steps, M: Model, λ: step-size, $\mathcal{L}$: Loss

for every 10 timesteps **do**
  $T \leftarrow$ noise ▷ Not heavy
  for each diffusion step $d = 1, \ldots, D$ **do**
    $T' \leftarrow M(T)$ ▷ Initial denoising, heavy
    for each guidance step $g = 1, \ldots, G$ **do**
      $G_T \leftarrow M(T')$ ▷ Denoise, heavy
      $T' \leftarrow T' - \lambda \nabla_{T'} \mathcal{L}(G_T)$ ▷ Backprop, heavy
    end for
  end for
end for

3.3 Safety-Critical Scenario Generation

The generation of realistic traffic scenarios in autonomous driving research typically relies on data-driven models trained on large-scale collections of real-world driving data. These models are effective at reproducing trajectories that remain within, or very close to, the observed data distribution. While this makes them well suited for capturing common driving behaviors, it is insufficient for generating safety-critical scenarios, as such events are inherently rare in naturalistic datasets. To address this limitation, alternative methods have been proposed that explicitly modify the trajectories of adversarial vehicles through optimization or perturbation techniques.

One straightforward approach is to directly identify safety-critical, or nearsafety-

critical, scenarios from recorded data [21, 22, 23]. Although this guarantees realistic scenarios, it is constrained by the sparsity of critical events in real-world data, often requiring computationally expensive searches across large datasets.

A second line of work focuses on optimizing the trajectory of a designated adversarial vehicle within an existing scenario [24, 25, 26]. In these approaches, the original trajectories are perturbed, commonly utilizing gradient descent, and optimized with respect to a predefined cost function. Although this often succeeds in generating safety-critical situations, the resulting scenarios frequently lack realism. In particular, adversarial agents may display implausible behaviors, such as systematically ignoring traffic rules or deviating from human driving patterns. Furthermore, the generated scenarios may either be unavoidable for the ego vehicle or otherwise fail to evaluate meaningful aspects of its decision-making algorithm. For instance, a rear-end collision that occurs while the ego-vehicle is stationary at a stop sign provides limited insights into its planning capabilities.

ReGents [26] partially addresses these shortcomings by employing a more sophisticated optimization objective. However, scenario diversity remains limited, being largely determined by hyperparameters within the objective function. Moreover, adversarial agents often focus on attacking the ego vehicle, while background traffic remains unresponsive to these changes, further reducing realism. Reinforcement-learning-based approaches that directly control adversarial agents face similar limitations [27].

3.3.1 Conditioned Scenario Generation

More recently, diffusion models [28] have emerged as a promising framework and have been widely adopted for traffic scenario generation [1, 7, 8, 9, 18]. These approaches learn to model the distribution of realistic traffic trajectories. This data-driven prior captures the statistical regularities of naturalistic traffic and ensures that generated scenarios remain plausible. During inference, the diffusion process can be guided toward specific outcomes by incorporating differentiable objectives, a strategy analogous to classifier guidance but without relying on an explicit classifier. Instead, the surrogate objectives directly shape agent behavior while maintaining consistency with the learned traffic prior.

Building on this principle, CTG [18] employs differentiable logic to specify target agent behaviors, whereas VBD [1] and DiffScene [8] integrate task-specific objectives as guidance signals. This guidance conditions the diffusion process toward specified outcomes while preserving plausibility under the learned traffic prior, enabling the generation of novel yet realistic scenarios.

In contrast, methods such as AdvDiffuser [7] employs a reinforcement-learning-inspired training procedure in which an adversarial network is trained to manipulate the entire scenario during the diffusion process. The resulting adversarial policies have been shown to be particularly effective against structurally similar planners, indicating a degree of overfitting to the planner architecture used in training. This observation underscores a broader limitation of adversarial testing: the learned be-

haviors are often tailored to the specific weaknesses of the training planner, raising concerns about their generalizability. To address this challenge, Ding et al. [29] introduce an adaptive sampling strategy that explores diverse regions of the scenario space and captures multiple levels of risk.

Finally, VBD [1] combines classifier-guidance-like optimization with a distance-based collision metric. Coupled with an iterative best-response algorithm, a pursuer and an adversary alternately perform gradient ascent and descent on a specified objective, leading to complex multi-agent interactions. While this enables the generation of highly interactive scenarios, it comes at the cost of increased computational demands and, in some cases, reduced realism.

4

Method

Due to the inherent bias in adversarial methods for critical scenario generation, this thesis will focus on agent guidance to specific target points. This allows additional manual control over generated scenarios and moves any bias to the target specification, allowing changes to how critical scenarios are created without requiring re-training of the generating method. Since previous methods have suffered computational issues, algorithmic changes are necessary to reduce inference time.

This section presents the use of Classifier-Free Guidance, together with adjustments to the training procedure to incorporate target information. Different strategies for integrating target data into VBD [1] are outlined, along with a data augmentation method for improving closed-loop generation and an optimization algorithm for increasing target accuracy. A validation method for measuring target guidance performance is also introduced. The chapter concludes with a summary of all models and their configurations used for training and evaluation.

4.1 Classifier-Free guidance

In the field of image generation, diffusion models using classifier-free guidance have been applied with success to increase realism, controllability and quality in generated images [5]. Our hypothesis is that classifier-free guidance will show similar results in our domain, giving motion planning models trained through imitation learning the innate ability to guide agents to a desired target position while still following learned traffic rules to a higher degree, with minimal additional computation. To the best of our knowledge, classifier-free guidance has not previously been applied to the problem of motion planning.

4.2 Target Guidance

The general idea behind target guidance is to define a point that should be part of an agent's trajectory at a specific time. This allows for control over the scenario, but any inherent bias will come from the method choosing target points, not the generating algorithm itself.

To allow inherent guidance of individual agents in the diffusion process, target information needs to be fed as input into the model. In addition, the training loop

needs to be adapted to facilitate the guidance. The target point $\mathbf{x}_{\text{target}}$ can either be represented as the target trajectories in the data, with $\{x, y, yaw, v_x, v_y, t\}$ to allow for more fine-grained control, or a simpler $\{x, y, t\}$.

4.2.1 CFG-Inspired Agent Guidance

During training, a target is sampled uniformly from an agent's ground-truth future trajectory. The target is encoded in the same manner as other spatial inputs, ensuring consistency within the model's input representation and coherent integration. Furthermore, the relative position between the agent and the target is encoded using the Fourier encoder and passed into the cross-attention layers of the denoising network (see Figure 4.3.3). This enables the model to effectively capture spatial relationships and directional dependencies, which are crucial for guiding the generated actions toward the target.

The training loop is modified to incorporate the sampled target point as an additional input. An agent's target is represented both as a global point, and relative to said agent. The relative representation is either a simple $\mathbf{x}_{\text{agent}}$ - $\mathbf{x}_{\text{target}}$, or the point transformed into the agents local coordinate system. These representations will be denoted as the relation function $\mathcal{R}$ in the algorithm below.

Algorithm 2 Target-Conditioned Training

Require: Initial state $\mathbf{x}_{\text{init}}$, Ground truth trajectory $\mathbf{x}_{\text{gt}}$, Probability p_{target}, Encoder E, Target Encoder T, Fourier Encoder F, Denoiser D, Relation function $\mathcal{R}$, Loss function $\mathcal{L}$, Model parameters θ

for each training instance **do**
  for each agent **do**
    if with probability p_{target} **then**
      $\mathbf{x}_{\text{target}} \sim \text{Uniform}(\mathbf{x}_{gt})$
      $\mathbf{z}_{\text{target}} \leftarrow T(\mathbf{x}_{\text{target}})$ ▷ Target encoding
      $\mathbf{r}_{\text{target}} \leftarrow F(\mathcal{R}(\mathbf{x}_{\text{agent}}, \mathbf{x}_{\text{target}}))$ ▷ Encoded relative pose
      $\hat{\mathbf{x}} \leftarrow D(E(\mathbf{x}_{\text{init}}, \mathbf{z}_{\text{target}}), \mathbf{r}_{\text{target}})$
    else
      $\hat{\mathbf{x}} \leftarrow D(E(\mathbf{x}_{\text{init}}))$
    end if
    Update θ using loss $\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x}_{gt})$
  end for
end for

This training setup is designed to encourage the model to interpret the target as a constraint embedded within a realistic, context-aware trajectory. By conditioning on a target point while requiring reconstruction of the ground truth sequence, the model is incentivized to generate trajectories that comply with the dynamics of the environment, including road structure, traffic rules, and social interactions.

The random sampling of target points within the trajectory allows for targets to not only be the final destination of a trajectory, but also representing a location within

the trajectory. This allows for early targets, but also allows for rollout towards a specific target using replanning, where said target gets closer for each planning iteration. The intention is for the model to learn how to generate goal-directed, yet behaviorally valid motion. At inference time, arbitrary target points can be specified to guide specific agents in the models output in a controllable and interpretable manner.

This training setup is designed to encourage the model to interpret the target as a constraint embedded within a realistic, context-aware trajectory. By conditioning on a target point while requiring reconstruction of the ground truth sequence, the model is incentivized to generate trajectories that comply with the dynamics of the environment, including road structure, traffic rules, and social interactions.

To promote flexibility, a target point is randomly sampled along the ground-truth trajectory, allowing it to serve as an intermediate waypoint, guiding the model toward a partial goal, at a specified timestep. This enables the model to handle iterative replanning, where the specified target moves progressively closer with each planning iteration. Ultimately, the goal is for the model to learn to produce goal-directed trajectories that remain behaviorally valid and dynamically consistent.

4.3 Implementing Classifier-Free Guidance

The VBD model uses two different types of encodings throughout the model. We have discussed these in section 2.6, where they are categorized as scene embeddings and relational embeddings. To match this embedding scheme, we chose to add embeddings to both the agents' scene embeddings and the their relational embeddings, this modification is shown in red in figure 4.3.1. Since the scene embeddings describe general information about the scene elements, we decided to add global target information to these embeddings. In a similar manner, relational embeddings of the targets are added to the agents' respective relational embeddings. Two different relational target embeddings are compared. The first type uses a relative frame of reference to the agent, subtracting the global agent pose from the global target. The second type encodes the target in the local frame of reference of the agent.

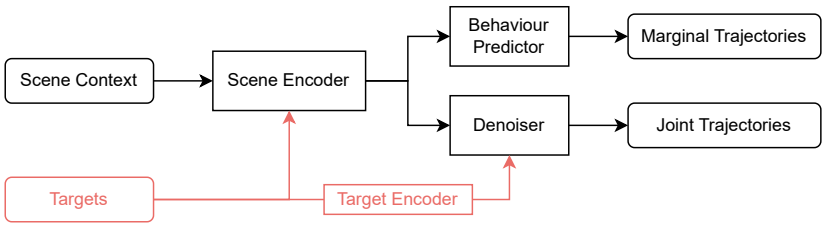


Figure 4.3.1: Overview of classifier-free guidance implementation.

Figure 4.3.2 shows a diagram of the modifications to the scene encoder. Global target data is encoded using a combination of an MLP, similar to all other scene

embeddings, and a time embedding. The embedding is then added to the agents' respective scene embeddings.

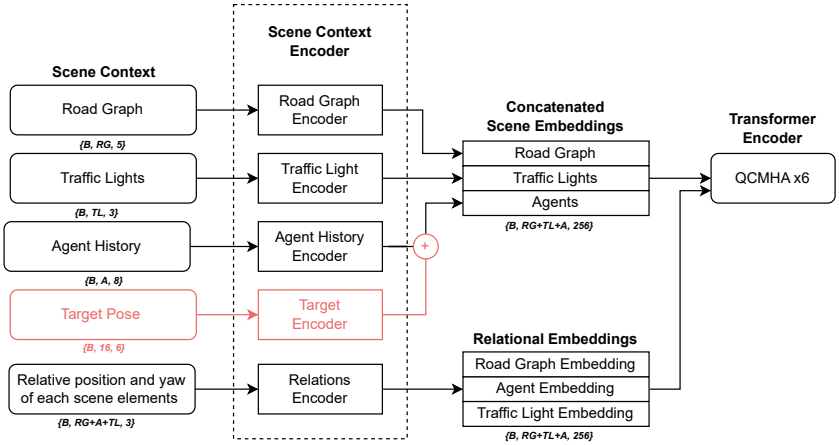


Figure 4.3.2: Scene Encoder for all CFG modifications.

4.3.1 Embedding configuration

Four configurations for incorporating the relational target embeddings into the denoiser were evaluated, including a baseline model without inserting the relational target embedding into the denoiser. The following table contain the four models that use the different denoiser implementations.

CFG: The embedding is injected at multiple points: into the query before the attention blocks and within each cross-attention layer.

CFG Attention Modification 1: The embedding is injected only in the cross-attention layers.

CFG Attention Modification 2: The embedding is injected into the query before the first block and into the first cross-attention layer.

CFG Encoder Only: No embedding is used (baseline).

These variations were designed to investigate the trade-off between guidance strength and general traffic generation performance. Configuration 1 maximizes the influence of the target embedding, but this can lead to overfitting: the model may over-prioritize the target and ignore external constraints such as road geometry. Moreover, repeated injection risks amplifying noise in trajectory refinement, resulting in degraded performance. Configuration 2 reduces this risk by restricting the embedding to the cross-attention layers, while Configuration 3 further limits its effect to the initial query and the first cross-attention layer, allowing the model to form a clean representation in later stages. Finally, Configuration 4 serves as a baseline, allowing us to isolate the effect of embedding injection by comparison with an unguided model.

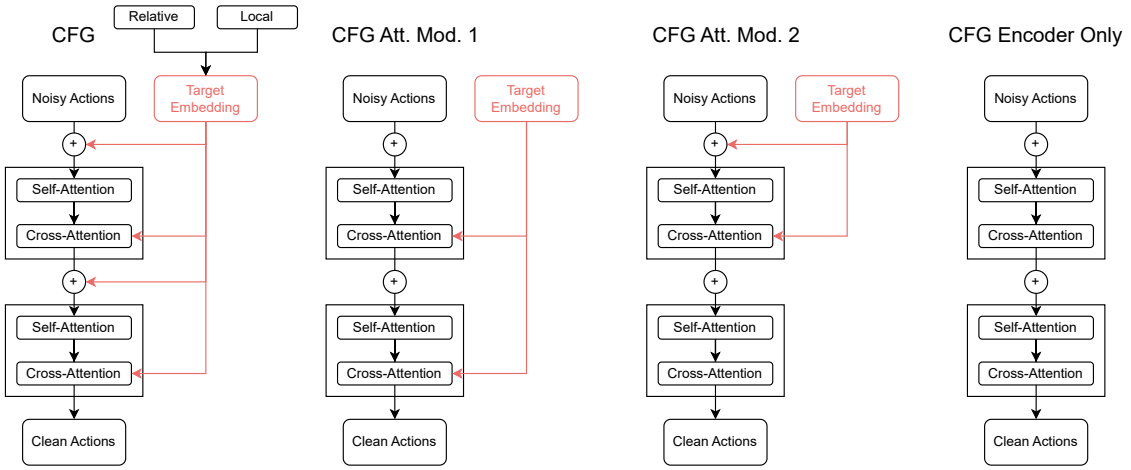
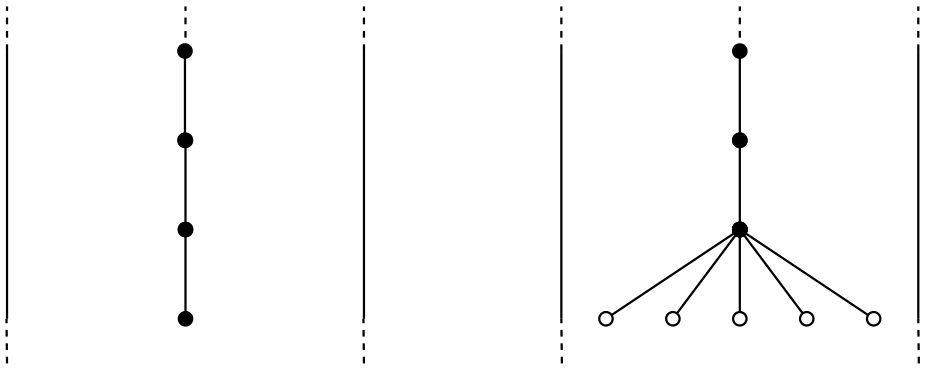


Figure 4.3.3: The four denoiser configurations tested

4.4 Perturbed Start Positions

To address the problem of compounding errors when using the model for close-loop generation, all start positions were perturbed perpendicular to their direction of travel, this is illustrated in figure 4.4.1. This data augmentation was designed to facilitate the models ability to self-correct when agents deviated from realistic trajectories, e.g. driving off-road. The perturbation was only applied to agents that moved more than 3 meters during the scenario, an arbitrary value used to avoid unwanted behavior from parked vehicles. The noise was sampled from a uniform distribution, with a maximum deviation of 1.2 meters. This distance was selected manually based on observations, in order to keep the agents close to their ground truth trajectories.



(a) Ground truth trajectory. (b) Start position selected at random.

Figure 4.4.1: Illustration of how the start position is changed in relation to the ground truth trajectory and the road markings.

4.5 Trajectory Optimization

To further improve the target accuracy of generated trajectories, we define an a simple algorithm for direct trajectory optimization. The algorithm operates directly on an agent's action sequence, adjusting the actions such that the resulting rolled-out trajectory is steered closer to the target. The updated action sequence is rolled out using a kinematic bicycle model. By keeping the step size small, the optimized trajectory preserves characteristics of the original while guiding the agent closer to the desired target. The optimization procedure consists of 1000 gradient descent steps with a step size of $1e{-}10$, minimizing the Euclidean distance to the target. Since the algorithm does not take any scene context into account, a key limitation with this method is that there are no guarantees that the agents will follow the road, nor reach the target using a realistic trajectory. This issue is minimized by keeping the number of steps, and step size, small. However, for the purpose of creating a safety-critical scenario, following road norms is not strictly necessary.

Optimizing step size and number of update steps for the tradeoff between target accuracy and realism has not been extensively explored. The current values allow for relatively small changes to already accurate trajectories, and larger changes to less accurate trajectories.

Algorithm 3 Trajectory Optimization via Action Sequence Adjustment

Require: initial action sequence $\mathbf{a}$, target position $\mathbf{x}_{\text{target}}$, Optimized timestep $\boldsymbol{t}$, kinematic bicycle model $\mathcal{M}$, step size $\eta = 10^{-10}$, number of steps $N = 1000$

for $i = 1$ to N **do**
  $\mathbf{x} \leftarrow \mathcal{M}(\mathbf{a})$ ▷ Roll out trajectory
  $\mathcal{L} = \|\mathbf{x}_{\boldsymbol{t}} - \mathbf{x}_{\text{target}}\|_2^2$
  $\mathbf{a} \leftarrow \mathbf{a} - \eta \nabla_{\mathbf{a}} \mathcal{L}$ ▷ Update actions
end for
$\mathbf{x}^* \leftarrow \mathcal{M}(\mathbf{a})$ ▷ Roll out optimized trajectory
return $\mathbf{x}^*$

4.6 Training and Model Configurations

To evaluate the different implementation methods for classifier-free guidance, the impact of the target point's reference frame and start position perturbations, we have trained 11 different models. Table 4.6.1 contain the different configurations and their name which is used in the rest of the thesis. Due to the long training time of a single model and the available compute, the number of configurations trained had to be limited.

All model were trained for 16 epochs each on the complete WOMD training dataset. Batch size 14 was selected to utilize all available GPU resources. The ADAM optimizer was used with an exponential rate going from $1e{-}3$ to $1e{-}6$ with a factor of x.

The second section of table 4.6.1 show three models trained with the original VBD implementation. Early evaluation suggested that using athe Log noise scheduler from the original VBD implementation performed better at closed-loop generation and is therefore the basis for all implementations of our classifier-free guidance models. The first model in the table known as *CFG*, is our base-model for classifier-free guidance and is the starting point for all other training configurations.

To read table 4.6.1, going from left to right, the columns shows: 1) the model name, 2) target point reference frame, either relative (Rel.) or local (Loc.), 3) start point perturbation (Ptrbd), 4) The denoiser configuration used: dense (d), attention modification 1 (1), and attention modification 2 (2). The attention modifications are shown in Figure 4.3.3.

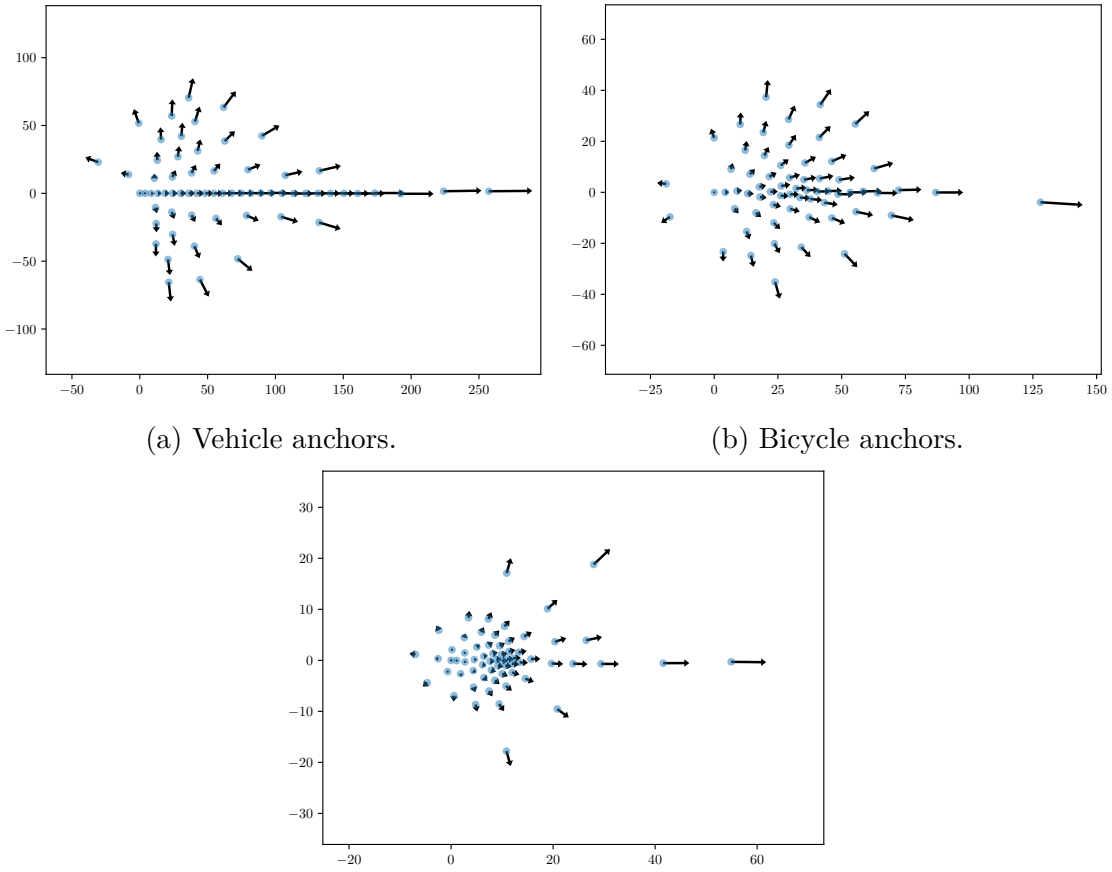
Model Name	Ref. Frame		Ptrbd	Denoiser		
	Rel.	Loc.		d	1	2
CFG	✓			✓		
CFG Perturb	✓		✓	✓		
CFG Local		✓		✓		
CFG Local Perturb		✓	✓	✓		
CFG att. mod. 1	✓				✓	
CFG att. mod. 2	✓					✓
CFG att. mod. 1 Local Perturb		✓	✓			
CFG No Target Loss	✓					
CFG Encoder Only	✓					
VBD						
VBD Perturb			✓			

Table 4.6.1: Model configurations.

4.7 Model Validation

The models were validated on 250 random scenarios from the WOMD validation dataset using the same metrics as in VBD [1]. Values were collected for metrics: ADE, collision, offroad, wrong-way and kinematic infeasibility. The models were both validated with a replan frequency of 10 and 80. This corresponds to close-loop generation updating the model every second and open-loop generation, generating trajectories only at the beginning of the scenario.

To compare the target guidance performance we created a new validation method. For unseen target points, the method uses random anchor points collected from the training dataset, shown in figure 4.7.1. For each scenario a single random agent is selected and given a random anchor point to reach, the anchor point are always on the road, making them plausible. All models received the same scenarios, guided agents and target points, to make the comparison as fair as possible. To measure target guidance performance, we record: minimum distance, final distance, yaw error and speed error. We also recorded the metrics for unguided scenarios mentioned previously to compare scenario quality of the background vehicles.



(a) Vehicle anchors.

(b) Bicycle anchors.

(c) Pedestrian anchors.

Figure 4.7.1: The three anchor point types.

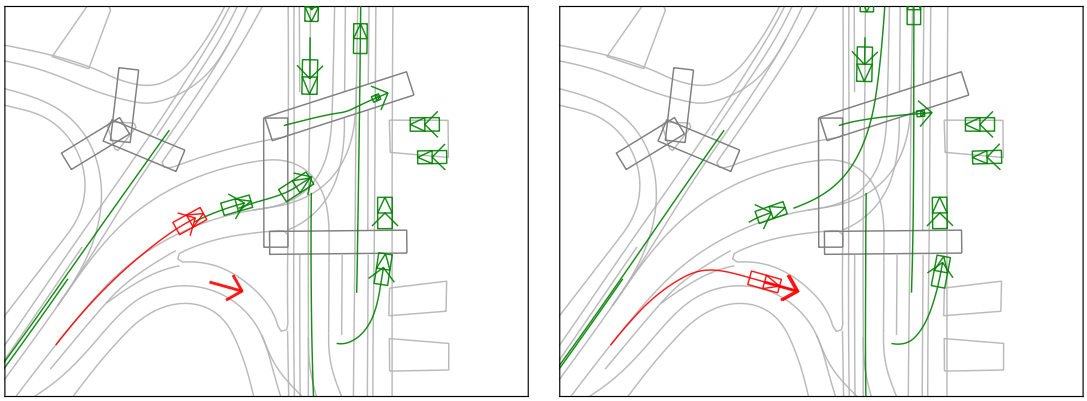
Target guidance is evaluated on all trained models with a total of nine different guidance configurations, not counting replan frequency. VBD guidance is compared with 1 and 10 guidance iteration steps, using the default step-size of 1.0. All combinations of VBD guidance, CFG guidance, and trajectory optimization are evaluated. The models are also evaluated without any guidance.

5

Results

5.1 Visual

To visualize scenarios, targets and generated trajectories we use the following type of images shown in figure 5.1.1. The gray lines are road markings, with crosswalks marked in a darker gray color. The images shows the end of an 8 second scenario and the green and red lines are the trajectories taken by the agents. An agent is represented by a rectangle with an inscribed triangle showing the heading of the agent. Agents also show their final speed indicated by the length of the arrow placed at the center of every agent. The bold red arrow seen in the center of figure 5.1.1a is the target of the agent marked in red. In figure 5.1.1b the agent is guided and reaches its target, while in figure 5.1.1a it was unguided. When guiding an agent, the objective is to align the position, speed and yaw of the agent with the target arrow.



(a) Agent not reaching target.

(b) Agent reaching target.

Figure 5.1.1: An example of a visualized scenario.

Figures shown in the following sections should be considered secondary to the results in our data tables, as generated trajectories can vary for the same scenario, model and guidance method. The figures are intended to illustrate the differences and similarities between the models and guidance methods and to highlight their strengths and weaknesses for the reader.

5.2 Runtime Comparison

To compare the runtime of each trajectory generation method with and without the guidance methods and using them with open-loop and closed-loop generation, we average the runtime of the same 10 scenarios for each method. Table 5.2.1 presents the average and standard deviation of the runtime for each separate trajectory generation method. Open-loop and closed-loop generation corresponds to replan 80 and replan 10. When closed-loop generation is used the same generation method is applied eight times in a row compared to open-loop generation which applies it once, which is consistent with the measured ratio.

VBD Guid.		**CFG Guid.**	**Traj. Opt.**	**Time (s)**	
Iter. 1	**Iter. 10**			**Replan 80**	**Replan 10**
				2.52 ± 0.179	19.9 ± 0.757
			✓	3.53 ± 0.193	27.7 ± 0.759
		✓		2.59 ± 0.205	20.5 ± 1
		✓	✓	3.52 ± 0.200	28.1 ± 0.760
	✓			50.6 ± 2.37	409 ± 14.7
	✓		✓	51.7 ± 1.74	409 ± 9.66
	✓	✓		50.7 ± 0.318	391 ± 4.55
	✓	✓	✓	51 ± 0.758	418 ± 7.59
✓				7.27 ± 0.134	58.6 ± 1.14
✓			✓	8.36 ± 0.154	66.2 ± 1.09
✓		✓		7.27 ± 0.135	57.8 ± 1.52
✓		✓	✓	8.38 ± 0.130	20.3 ± 0.698

Table 5.2.1: Empirical mean runtime with standard deviation for all guidance methods and trajectory optimization.

Table 5.2.1 shows that CFG guidance does not significantly increase runtime when used. VBD guidance, with 1 and 10 guidance iteration steps, increases the runtime by a factor of 2.9 and 20 compared to not using guidance. Combining CFG guidance with VBD guidance does not increase the runtime further. The fine-tuning step trajectory optimization adds one second to the runtime for all methods when used with open-loop generation and eight seconds with closed-loop generation, since it is applied at the end of each generation step.

5.3 Relative and Local Targets

To determine the effect on scenario quality and controllability when using targets encoded using a relative frame of reference compared to a local frame of reference, two models have been implemented: CFG and CFG Local. The original VBD model is used as a comparative baseline where it is applicable.

Comparing scenario quality metrics for the unguided trajectories, seen in table 5.3.1, shows an incurred penalty on both models, with increased scores, compared to the

original VBD model. For unguided trajectories there is no obvious preference for either target encoding method. Figure 5.3.1 shows unguided trajectories generated with the three models.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.762	1.01	0.877	0	0.0436	0.0599	0	0.0165	0.0361	0	$9.44e{-3}$	0.0387
CFG	0.885	1.23	1.55	0.0323	0.0564	0.0726	0	0.0286	0.0621	0	0.0116	0.0419
CFG Local	0.959	1.22	1.13	0.0323	0.0550	0.0711	0	0.0256	0.0492	0	0.0131	0.0458

Table 5.3.1: Scenario quality metrics of background agents for relative target and local target models with replan 10 and no guidance.

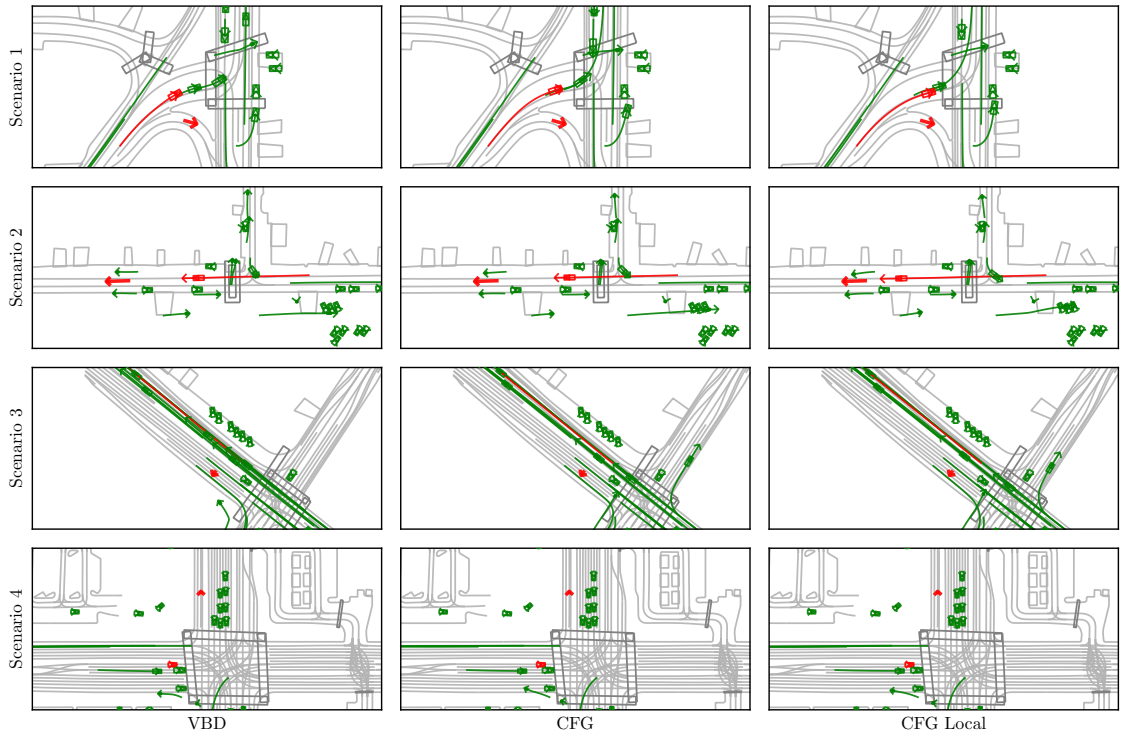


Figure 5.3.1: Scenarios with trajectories generated using relative target and local target models with replan 10 and no guidance.

Using VBD guidance, the results are similar to unguided generation, with CFG and CFG Local performing worse than the VBD baseline on all scenario quality metrics. These results are shown in table 5.3.2. The CFG model performs slightly better than CFG Local on the ADE metric, but incurs more collisions and off road driving. The target scores, shown in table 5.3.3, are greatly improved for both CFG and CFG Local compared to the VBD baseline, with CFG Local seeing the largest improvement.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.900	1.19	1.39	0.0445	0.0641	0.0760	0	0.0224	0.0470	0	0.0105	0.0313
CFG	0.999	1.24	1.50	0.0600	0.0735	0.0787	0	0.0306	0.0528	0	0.0109	0.0406
CFG Local	1.07	1.37	1.60	0.0645	0.0757	0.0846	0	0.0305	0.0581	0	0.0140	0.0445

Table 5.3.2: Scenario quality metrics of background agents for relative target and local target models with replan 10 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	2.16	15.8	24	5.70	16.8	23.6	1.25	3.01	3.51	0.458	0.813	0.862
CFG	1.54	14.1	23.2	3.24	15	23.1	1.01	2.77	3.29	0.504	0.837	0.878
CFG Local	1.15	12.8	21.4	3.96	13.7	21.2	1	2.55	3.10	0.521	0.810	0.837

Table 5.3.3: Target metrics for relative target and local target models with replan 10 and VBD guidance.

In figure 5.3.2 scenario 1, 3, and 4 illustrate the difference between the models best, with CFG Local being more prone to come closest to the targets compared to CFG. We also see that the VBD model often fails, accounting for the difference in scores.

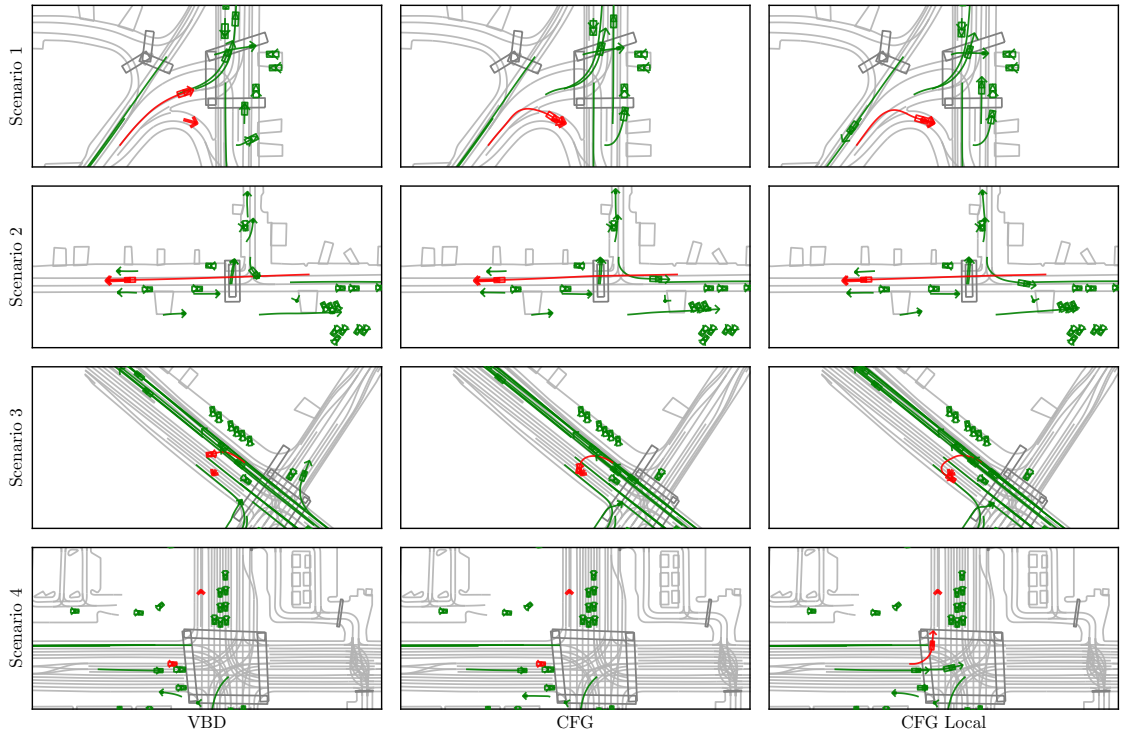


Figure 5.3.2: Scenarios with trajectories generated using relative target and local target models with replan 10 and VBD guidance.

For CFG guidance, using relative targets results in better scenario quality scores compared to local targets, as seen in table 5.3.4. However, target scores, seen in table 5.3.5, differ greatly between the two models, with CFG Local showing a much lower minimum distance and final distance compared to the CFG model. This highlights

a key difference between the two target encoding types, where local targets appear to be easier to learn for the model compared to a relative encoding.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.857	1.12	1.39	0.0645	0.0815	0.0881	0	0.0286	0.0529	0	0.0124	0.0436
CFG Local	0.906	1.17	1.38	0.0645	0.0846	0.0833	0	0.0301	0.0577	0	0.0141	0.0459

Table 5.3.4: Scenario quality metrics of background agents for relative target and local target models with replan 10 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	5.61	12.4	16.7	13.8	19.1	18.8	1.06	2.17	3.02	0.404	0.747	0.819
CFG Local	1.19	9.91	14.9	9.13	13.4	13.6	2.05	2.67	2.46	0.653	0.947	0.926

Table 5.3.5: Target metrics for relative target and local target models with replan 10 and CFG guidance.

Figure 5.3.3 shows a comparison between the two models when used with CFG guidance. The CFG model exhibits overshooting behavior which does not appear in the CFG Local model. This shows a clear difference between the effect of using the different target encodings in our models.

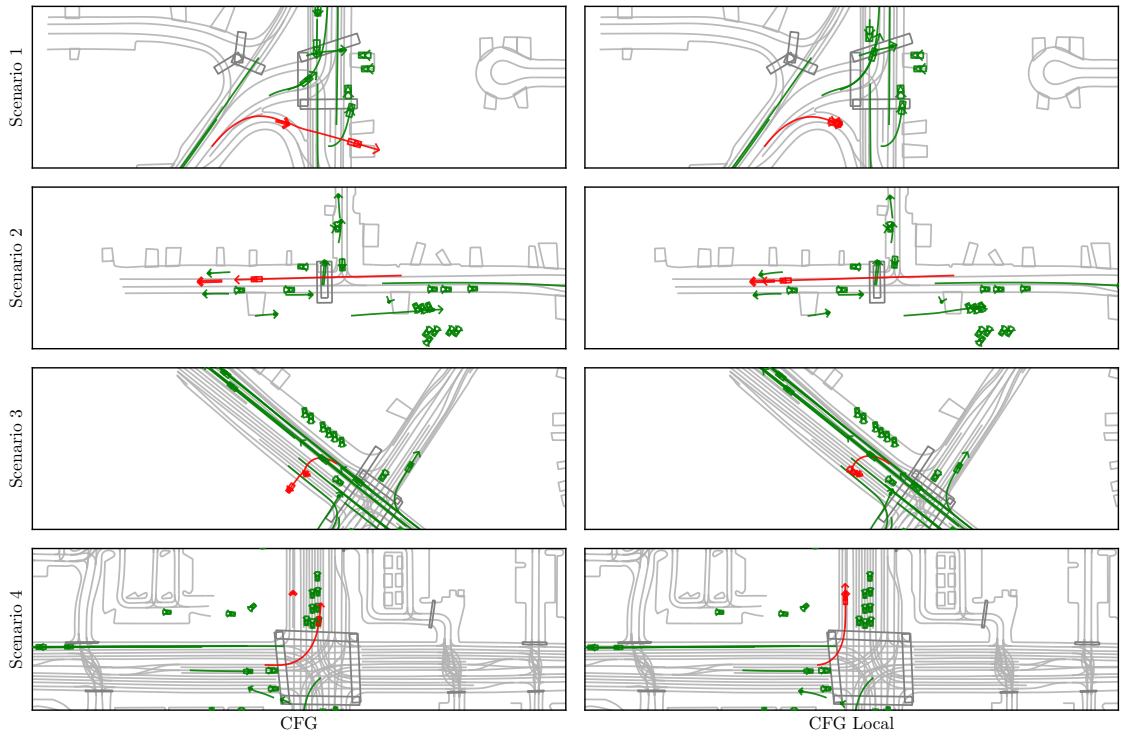


Figure 5.3.3: Scenarios with trajectories generated using relative target and local target models with replan 10 and CFG guidance.

When used with the combined CFG and VBD guidance method, relative targets result in better scenario quality scores compared to local targets, shown in table

5.3.6. However, local targets surpass relative targets for minimum and final distance, at the cost of a slight increase in the speed error and yaw error scores.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.933	1.09	1.42	0.0645	0.0871	0.0884	0	0.0285	0.0490	0	0.0107	0.0431
CFG Local	1.03	1.14	1.48	0.0645	0.0890	0.0894	0	0.0297	0.0565	0	0.0140	0.0447

Table 5.3.6: Scenario quality metrics of background agents for relative target and local target models with replan 10 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.542	2.66	6.25	0.713	3.12	6.39	0.178	0.927	1.65	0.104	0.373	0.672
CFG Local	0.294	1.74	5.07	0.490	2.23	5.13	0.361	1.08	1.83	0.186	0.420	0.626

Table 5.3.7: Target metrics for relative target and local target models with replan 10 and CFG and VBD guidance.

When combining both guidance methods, the overshooting behavior disappears from the CFG model, shown in figure 5.3.4.

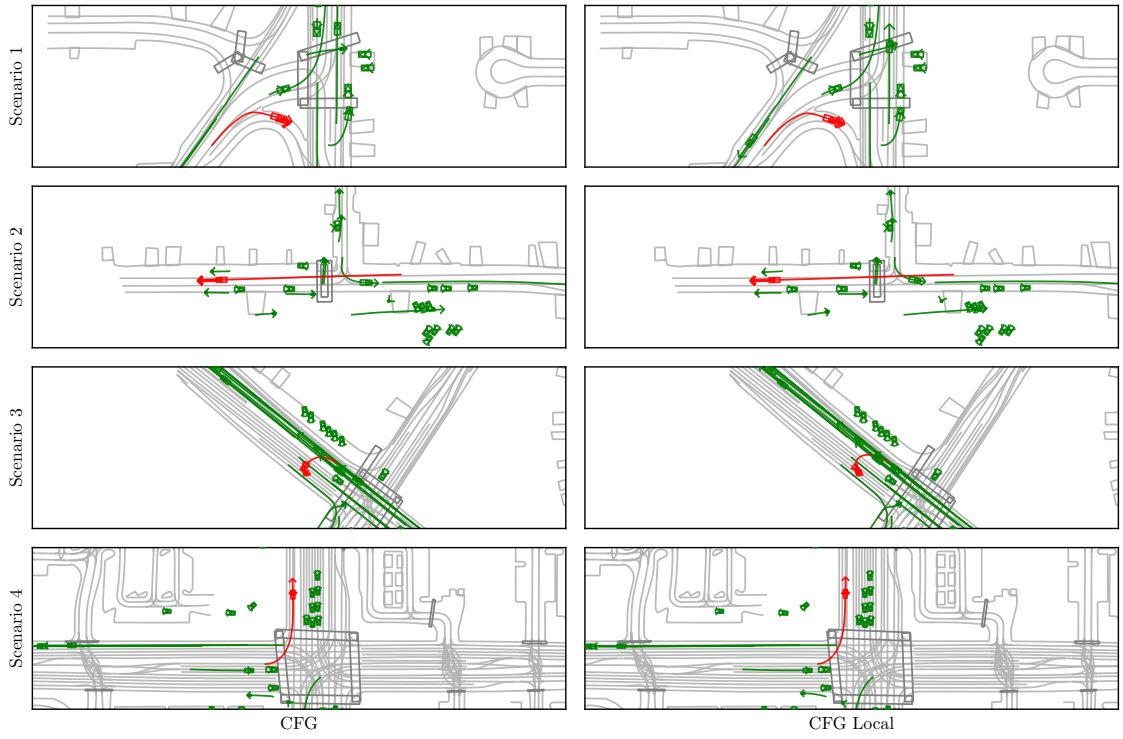


Figure 5.3.4: Scenarios with trajectories generated using relative target and local target models with replan 10 and CFG and VBD guidance.

5.4 Start Position Perturbation

To evaluate the impact on performance from training models with start position perturbation, a new data augmentation technique, three baseline models have been

used: VBD, CFG and CFG Local. Their generated trajectories are compared using: no guidance; VBD guidance with 10 guidance iteration steps; CFG guidance; and the combined CFG and VBD guidance method with 10 guidance iteration steps. All trajectories are generated with replan 10. The goal of this data augmentation is to improve the scenario quality metrics for closed-loop trajectory generation.

Scenario quality metrics for unguided trajectories, shown in table 5.4.1, are improved by the data augmentation technique for all models with the exception of the off road score of CFG Local, which is increased. However, this model shows the largest improvement in ADE, as well as a notable improvement in collision.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.762	1.01	0.877	0	0.0436	0.0599	0	0.0165	0.0361	0	$9.44e{-3}$	0.0387
VBD Perturb	0.782	1.01	0.901	0	0.0417	0.0574	0	0.0139	0.0351	0	$9.55e{-3}$	0.0400
CFG	0.885	1.23	1.55	0.0323	0.0564	0.0726	0	0.0286	0.0621	0	0.0116	0.0419
CFG Perturb	0.877	1.17	1.36	0	0.0459	0.0669	0	0.0239	0.0548	0	$7.39e{-3}$	0.0239
CFG Local	0.959	1.22	1.13	0.0323	0.0550	0.0711	0	0.0256	0.0492	0	0.0131	0.0458
CFG Local Perturb	0.848	1.14	0.984	0	0.0500	0.0693	0	0.0293	0.0503	0	0.0116	0.0426

Table 5.4.1: Scenario quality metrics of background agents for baseline models and data augmented models with replan 10 and no guidance.

Figure 5.4.1 shows examples of generated trajectories using both the baseline models and the data augmented models. We observe no clear difference between the included scenarios. Note that the generated scenarios are unguided, thus ignoring the target arrow.

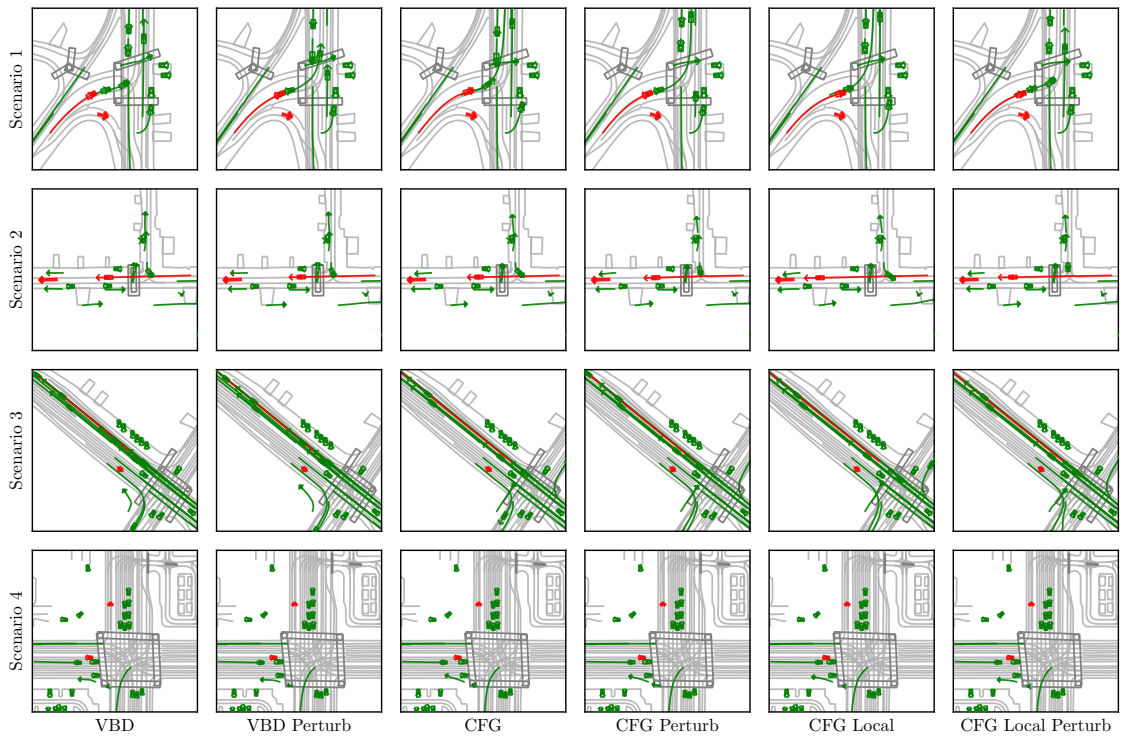


Figure 5.4.1: Scenarios with trajectories generated using baseline models and data augmented models with replan 10 and no guidance.

Next, we examine the metrics for trajectories generated using VBD guidance. Table 5.4.2 shows improved scenario quality among all models, with VBD showing the least improvement. In table 5.4.3 VBD show improved target scores, while both CFG and CFG Local have impaired scores. This suggests there is a trade-off between using the augmentation and target scores. Figure 5.4.2 show the generated trajectories.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.900	1.19	1.39	0.0445	0.0641	0.0760	0	0.0224	0.0470	0	0.0105	0.0313
VBD Perturb	0.938	1.17	1.43	0.0323	0.0611	0.0704	0	0.0209	0.0484	0	$9.71e{-3}$	0.0260
CFG	0.999	1.24	1.50	0.0600	0.0735	0.0787	0	0.0306	0.0528	0	0.0109	0.0406
CFG Perturb	0.932	1.15	1.26	0.0323	0.0649	0.0779	0	0.0237	0.0487	0	$9.51e{-3}$	0.0268
CFG Local	1.07	1.37	1.60	0.0645	0.0757	0.0846	0	0.0305	0.0581	0	0.0140	0.0445
CFG Local Perturb	0.983	1.21	1.46	0.0328	0.0663	0.0786	0	0.0308	0.0511	0	0.0125	0.0456

Table 5.4.2: Scenario quality metrics of background agents for baseline models and data augmented models with replan 10 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	2.16	15.8	24	5.70	16.8	23.6	1.25	3.01	3.51	0.458	0.813	0.862
VBD Perturb	1.60	14.9	24.2	2.69	15.5	24	1.09	2.89	3.50	0.521	0.850	0.875
CFG	1.54	14.1	23.2	3.24	15	23.1	1.01	2.77	3.29	0.504	0.837	0.878
CFG Perturb	1.92	15.1	23.9	3.39	15.8	23.8	1.34	3.15	3.50	0.456	0.896	0.914
CFG Local	1.15	12.8	21.4	3.96	13.7	21.2	1	2.55	3.10	0.521	0.810	0.837
CFG Local Perturb	1.43	14.5	24	2.81	15.2	23.9	1.37	3.02	3.36	0.491	0.898	0.897

Table 5.4.3: Target metrics for baseline models and data augmented models with replan 10 and VBD guidance.

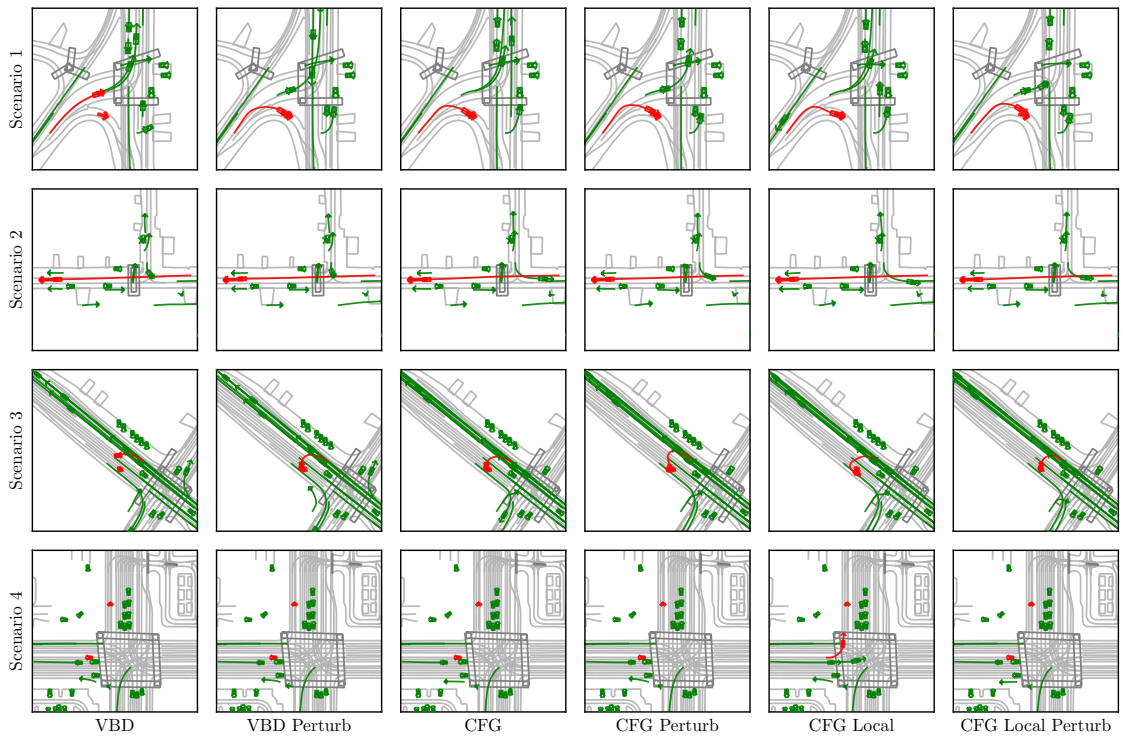


Figure 5.4.2: Scenarios with trajectories generated using baseline models and data augmented models with replan 10 and VBD guidance.

With CFG guidance we observe improvements for both scenario quality metrics and target metrics for the applicable models: CFG and CFG Local, when the augmentation is applied. The metrics are shown in tables 5.4.4 and 5.4.5. This highlights that the augmentation can be beneficial for target guidance, in contrast to what we saw with VBD guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.857	1.12	1.39	0.0645	0.0815	0.0881	0	0.0286	0.0529	0	0.0124	0.0436
CFG Perturb	0.893	1.08	1.23	0.0645	0.0734	0.0791	0	0.0248	0.0493	0	$7.66e{-3}$	0.0250
CFG Local	0.906	1.17	1.38	0.0645	0.0846	0.0833	0	0.0301	0.0577	0	0.0141	0.0459
CFG Local Perturb	0.909	1.07	1.25	0.0645	0.0727	0.0767	0	0.0326	0.0586	0	0.0134	0.0448

Table 5.4.4: Scenario quality metrics of background agents for baseline models and data augmented models with replan 10 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	5.61	12.4	16.7	13.8	19.1	18.8	1.06	2.17	3.02	0.404	0.747	0.819
CFG Perturb	4.85	11.9	15.9	14.4	18.6	17.4	1.15	2.29	3.33	0.419	0.758	0.824
CFG Local	1.19	9.91	14.9	9.13	13.4	13.6	2.05	2.67	2.46	0.653	0.947	0.926
CFG Local Perturb	1.09	8.92	13.2	7	11.9	12.3	1.92	2.66	2.47	0.637	0.951	0.909

Table 5.4.5: Target metrics for baseline models and data augmented models with replan 10 and CFG guidance.

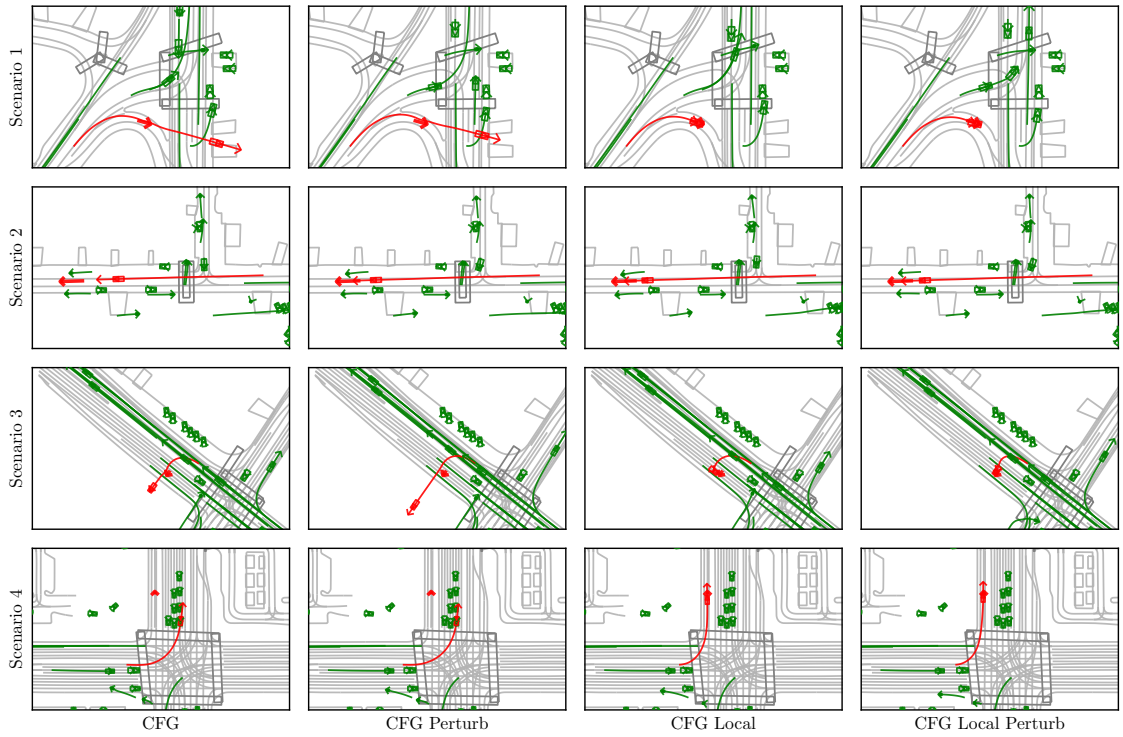


Figure 5.4.3: Scenarios with trajectories generated using baseline models and data augmented models with replan 10 and CFG guidance.

The combined CFG guidance and VBD guidance method show improvement for the scenario quality scores for the augmented models, shown in table 5.4.6. Target

scores, shown in table 5.4.7, are only improved for the CFG model, while the CFG Local model sees a reduction in performance. The main reduction is seen in the minimum distance, while the final distance remain the same.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.933	1.09	1.42	0.0645	0.0871	0.0884	0	0.0285	0.0490	0	0.0107	0.0431
CFG Perturb	0.867	1.05	1.35	0.0645	0.0814	0.0816	0	0.0219	0.0353	0	$8.31e{-3}$	0.0267
CFG Local	1.03	1.14	1.48	0.0645	0.0890	0.0894	0	0.0297	0.0565	0	0.0140	0.0447
CFG Local Perturb	0.969	1.05	1.36	0.0645	0.0763	0.0807	0	0.0283	0.0413	0	0.0142	0.0458

Table 5.4.6: Scenario quality metrics of background agents for baseline models and data augmented models with replan 10 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.542	2.66	6.25	0.713	3.12	6.39	0.178	0.927	1.65	0.104	0.373	0.672
CFG Perturb	0.506	2.17	4.92	0.660	2.74	5.32	0.186	0.991	1.77	0.0988	0.388	0.697
CFG Local	0.294	1.74	5.07	0.490	2.23	5.13	0.361	1.08	1.83	0.186	0.420	0.626
CFG Local Perturb	0.404	1.82	5.16	0.640	2.33	5.30	0.360	1.06	1.66	0.163	0.459	0.703

Table 5.4.7: Target metrics for baseline models and data augmented models with replan 10 and CFG and VBD guidance.

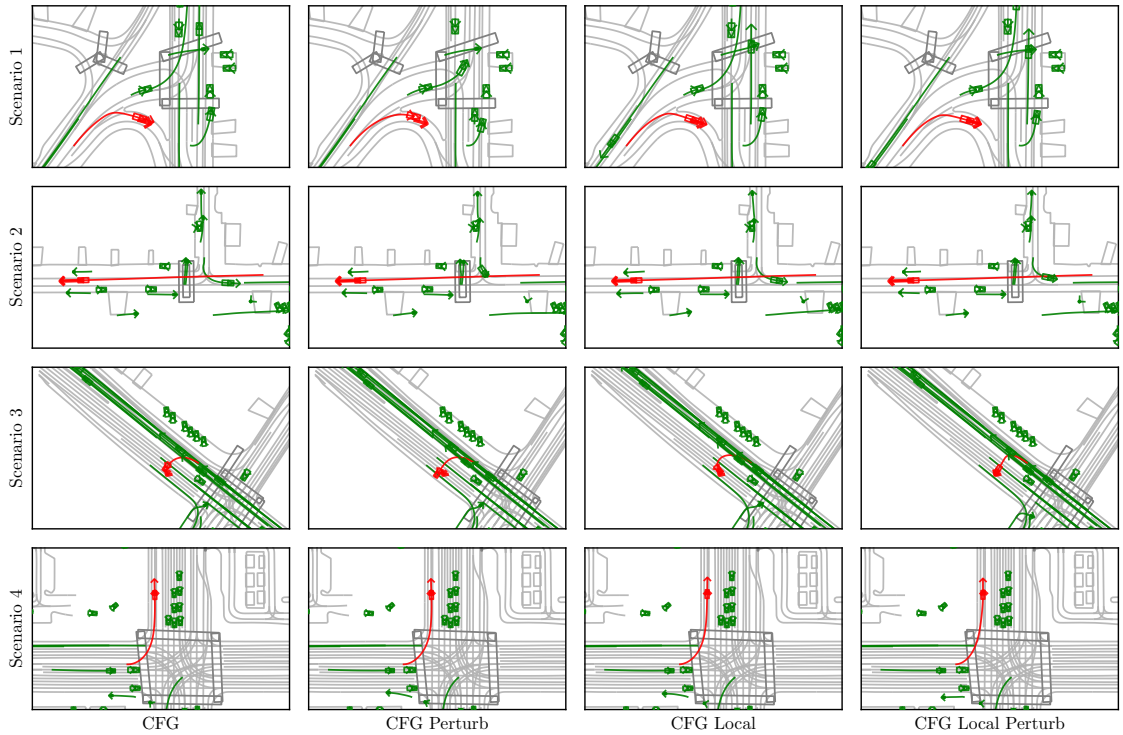


Figure 5.4.4: Scenarios with trajectories generated using baseline models and data augmented models with replan 10 and CFG and VBD guidance.

5.5 Denoiser Implementations

We evaluated four denoiser implementations corresponding to the models: CFG, CFG Attention Modification 1, CFG Attention Modification 2 and CFG Encoder Only. All models use relative targets and they differ in how the target data is incorporated in the denoiser. All models except CFG Encoder Only uses a modified denoiser. For a detailed explanation of the differences between the implementations see section 4.3.1.

Table 5.5.1 shows the scenario quality scores of the unguided trajectories from the four models and the original VBD model, included as a baseline. The four models show slight variations for the scenario quality scores, with attention modification 1 and 2 having the best performance. All models perform worse than the original VBD model.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.762	1.01	0.877	0	0.0436	0.0599	0	0.0165	0.0361	0	$9.44e{-3}$	0.0387
CFG	0.885	1.23	1.55	0.0323	0.0564	0.0726	0	0.0286	0.0621	0	0.0116	0.0419
CFG att. mod. 1	0.844	1.11	0.958	0	0.0529	0.0722	0	0.0301	0.0755	0	0.0107	0.0330
CFG att. mod. 2	0.891	1.12	0.932	0	0.0511	0.0675	0	0.0323	0.0762	0	0.0113	0.0418
CFG Enc. only	0.878	1.25	1.57	0.0323	0.0578	0.0744	0	0.0302	0.0611	0	0.0118	0.0422

Table 5.5.1: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 10 and no guidance.

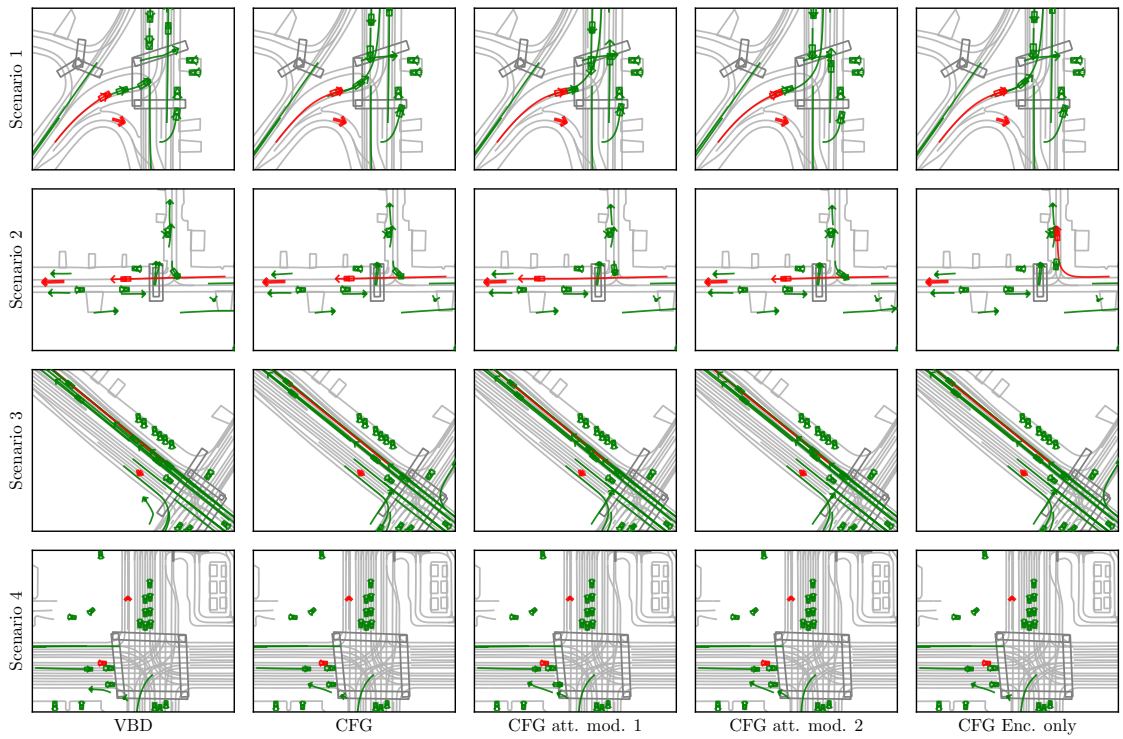


Figure 5.5.1: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 10 and no guidance.

Using VBD guidance, scenario quality metrics are again worse than the VBD baseline.

Scenario quality scores show slight variation between the four models when used with VBD guidance. CFG perfoms the best, while CFG Att. Mod. 2 have the worst scores. Target metric scores show CFG Att. Mod. 2 and CFG Att. Mod. 1 performing best. CFG and CFG Encoder Only perform similarly.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.900	1.19	1.39	0.0445	0.0641	0.0760	0	0.0224	0.0470	0	0.0105	0.0313
CFG	0.999	1.24	1.50	0.0600	0.0735	0.0787	0	0.0306	0.0528	0	0.0109	0.0406
CFG att. mod. 1	1.07	1.33	1.40	0.0323	0.0632	0.0760	0	0.0300	0.0505	0	0.0117	0.0324
CFG att. mod. 2	1.09	1.36	1.48	0.0488	0.0712	0.0841	0.0323	0.0387	0.0591	0	0.0174	0.0521
CFG Enc. only	1.03	1.28	1.67	0.0645	0.0770	0.0823	0	0.0318	0.0561	0	0.0124	0.0425

Table 5.5.2: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 10 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	2.16	15.8	24	5.70	16.8	23.6	1.25	3.01	3.51	0.458	0.813	0.862
CFG	1.54	14.1	23.2	3.24	15	23.1	1.01	2.77	3.29	0.504	0.837	0.878
CFG att. mod. 1	1.23	13.7	23.1	2.90	14.5	23.1	0.940	2.73	3.41	0.500	0.808	0.840
CFG att. mod. 2	1.30	13.1	21.7	2.71	14.1	21.7	0.883	2.57	3.24	0.467	0.787	0.845
CFG Enc. only	1.36	14.4	23.5	3.27	15.3	23.3	1.06	2.81	3.32	0.484	0.846	0.896

Table 5.5.3: Target metrics for the four models using different denoiser implementations with replan 10 and VBD guidance.

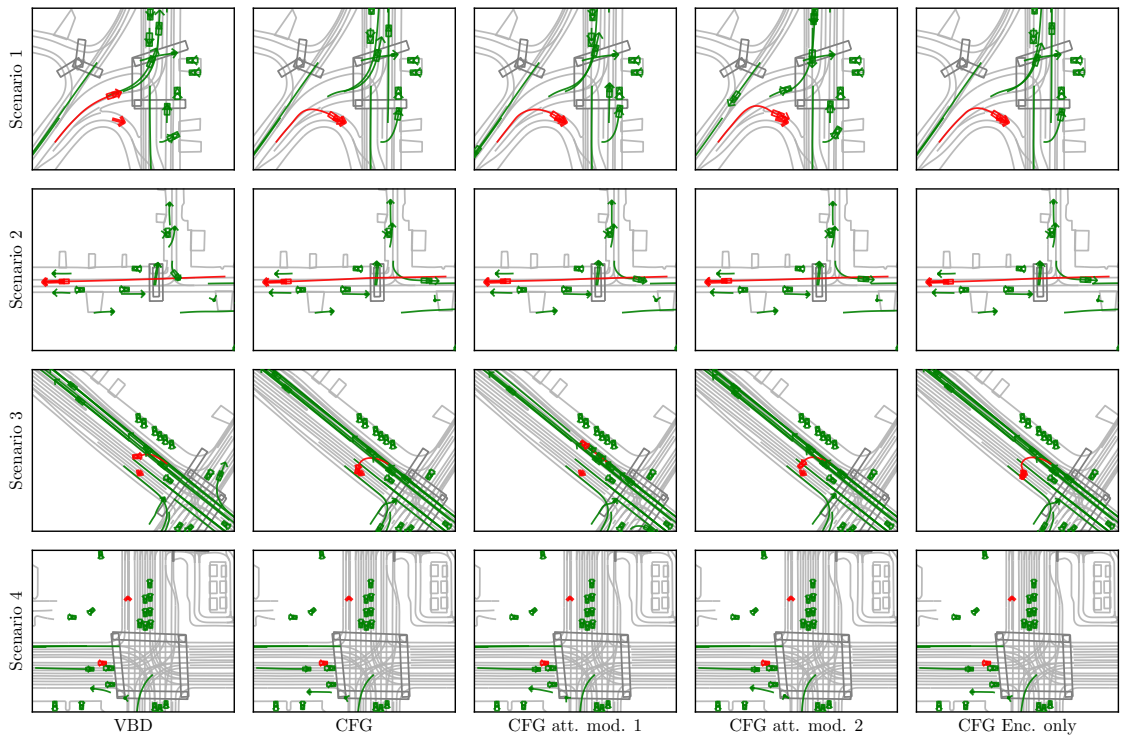


Figure 5.5.2: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 10 and VBD guidance.

Using CFG guidance, CFG Att. Mod. 1 and CFG Att. Mod. 2 show a slight advantage over the other two models. The modified models include a slight improvement in both scenario quality and target metrics.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.857	1.12	1.39	0.0645	0.0815	0.0881	0	0.0286	0.0529	0	0.0124	0.0436
CFG att. mod. 1	0.890	1.07	1.25	0.0645	0.0780	0.0894	0	0.0328	0.0767	0	0.0122	0.0366
CFG att. mod. 2	0.888	1.06	1.21	0.0645	0.0766	0.0836	0	0.0352	0.0769	0	0.0118	0.0430
CFG Enc. only	0.839	1.14	1.41	0.0645	0.0845	0.0906	0	0.0314	0.0757	0	0.0106	0.0287

Table 5.5.4: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 10 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	5.61	12.4	16.7	13.8	19.1	18.8	1.06	2.17	3.02	0.404	0.747	0.819
CFG att. mod. 1	5.63	11.6	15.4	13.5	18.6	18.8	0.914	1.94	3.04	0.395	0.734	0.837
CFG att. mod. 2	4.54	11.6	15.7	14.4	19.2	18.9	1.01	2.35	3.60	0.379	0.710	0.800
CFG Enc. only	5.66	12.4	16.8	13.7	19.1	18.8	1.01	2.15	2.98	0.426	0.740	0.792

Table 5.5.5: Target metrics for the four models using different denoiser implementations with replan 10 and CFG guidance.

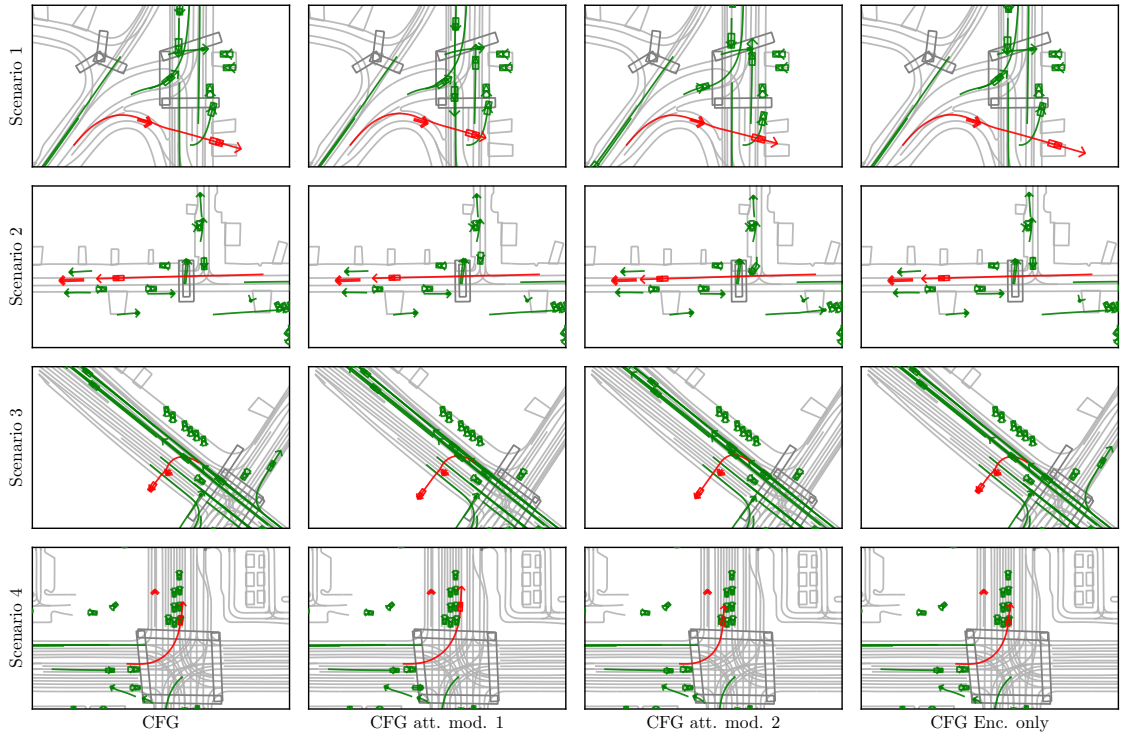


Figure 5.5.3: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 10 and CFG guidance.

For the combined CFG and VBD guidance method, CFG Att. Mod. 1 performs best out of all models on the scenario quality metrics, while CFG Att. Mod. 2 have the best target metrics scores.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.933	1.09	1.42	0.0645	0.0871	0.0884	0	0.0285	0.0490	0	0.0107	0.0431
CFG att. mod. 1	0.889	1.01	1.40	0.0645	0.0806	0.0797	0	0.0327	0.0606	0	0.0115	0.0342
CFG att. mod. 2	0.909	1.07	1.43	0.0645	0.0842	0.0892	0.0323	0.0387	0.0627	0	0.0152	0.0449
CFG Enc. only	0.913	1.12	1.46	0.0645	0.0873	0.0859	0	0.0307	0.0512	0	0.0114	0.0433

Table 5.5.6: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 10 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.542	2.66	6.25	0.713	3.12	6.39	0.178	0.927	1.65	0.104	0.373	0.672
CFG att. mod. 1	0.584	2.62	5.97	0.874	3.48	6.37	0.198	0.976	1.66	0.0951	0.422	0.743
CFG att. mod. 2	0.471	1.66	3.80	0.725	2.90	5	0.205	0.868	1.41	0.106	0.413	0.717
CFG Enc. only	0.537	2.68	6.28	0.703	3.18	6.60	0.184	0.905	1.62	0.108	0.374	0.670

Table 5.5.7: Target metrics for the four models using different denoiser implementations with replan 10 and CFG and VBD guidance.

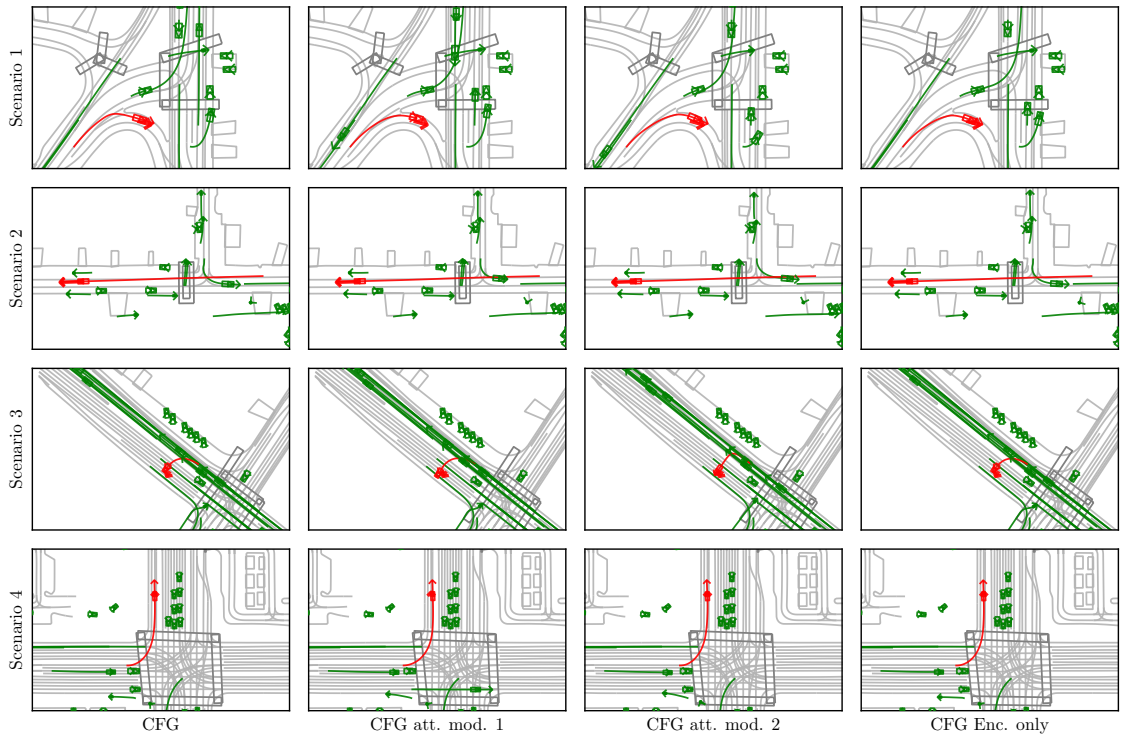


Figure 5.5.4: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 10 and CFG and VBD guidance.

5.6 Final Model

We propose a final model that combines the design choices of the best performing models analyzed in the previous sections, to improve the performance further. The proposed model uses start position perturbation, because of the observed improvement for guided, closed-loop trajectory generation with respect to both scenario quality and target metrics with the CFG guidance method. We use targets encoded in the agents local frame of reference since they improve target metrics scores and reduces the overshooting problem. For the denoiser implementation, Attention Modification 1 is selected for the improved scenario quality seen in Table 5.5.6 while preserving low target scores.

The final model is compared against CFG Local Perturb and CFG Attention Modification 1, as baseline models. Without guidance, the scenario quality scores, shown in table 5.6.1, are all improved with the final model except for the off road score.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	0.959	1.22	1.13	0.0323	0.0550	0.0711	0	0.0256	0.0492	0	0.0131	0.0458
CFG Local Perturb	0.848	1.14	0.984	0	0.0500	0.0693	0	0.0293	0.0503	0	0.0116	0.0426
CFG att. mod. 1	0.844	1.11	0.958	0	0.0529	0.0722	0	0.0301	0.0755	0	0.0107	0.0330
CFG att. mod. 1 Local Perturb	0.825	1.11	0.987	0	0.0464	0.0651	0	0.0327	0.0560	0	$8.84e{-3}$	0.0271

Table 5.6.1: Scenario quality metrics of background agents for the final model and its predecessors with replan 10 and no guidance.

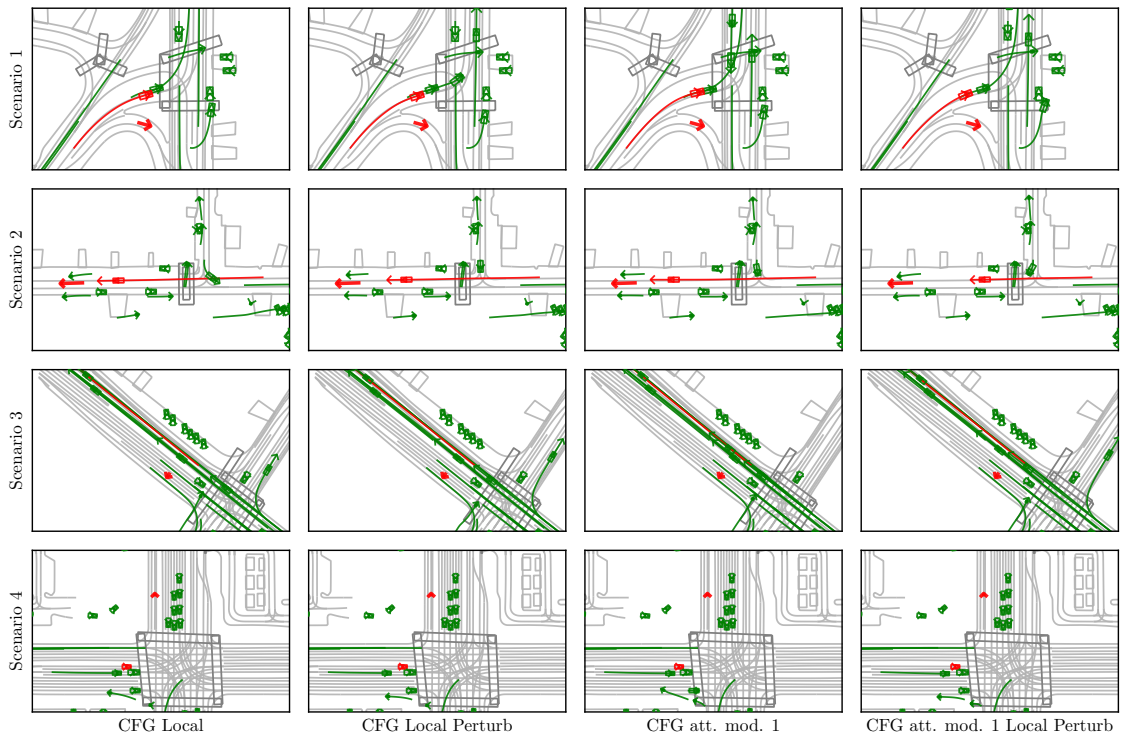


Figure 5.6.1: Scenarios with trajectories generated using the final model and its predecessors with replan 10 and no guidance.

Using VBD guidance the final model performs similarly to CFG Local Perturb on the scenario quality metrics. The target metrics are similar to CFG attention modification 1.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.07	1.37	1.60	0.0645	0.0757	0.0846	0	0.0305	0.0581	0	0.0140	0.0445
CFG Local Perturb	0.983	1.21	1.46	0.0328	0.0663	0.0786	0	0.0308	0.0511	0	0.0125	0.0456
CFG att. mod. 1	1.07	1.33	1.40	0.0323	0.0632	0.0760	0	0.0300	0.0505	0	0.0117	0.0324
CFG att. mod. 1 Local Perturb	0.991	1.25	1.34	0.0442	0.0648	0.0739	0	0.0322	0.0622	0	$9.40e{-3}$	0.0319

Table 5.6.2: Scenario quality metrics of background agents for the final model and its predecessors with replan 10 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.15	12.8	21.4	3.96	13.7	21.2	1	2.55	3.10	0.521	0.810	0.837
CFG Local Perturb	1.43	14.5	24	2.81	15.2	23.9	1.37	3.02	3.36	0.491	0.898	0.897
CFG att. mod. 1	1.23	13.7	23.1	2.90	14.5	23.1	0.940	2.73	3.41	0.500	0.808	0.840
CFG att. mod. 1 Local Perturb	1.20	13.9	23.1	3	14.8	23	1.04	2.83	3.38	0.542	0.879	0.884

Table 5.6.3: Target metrics for the final model and its predecessors with replan 10 and VBD guidance.

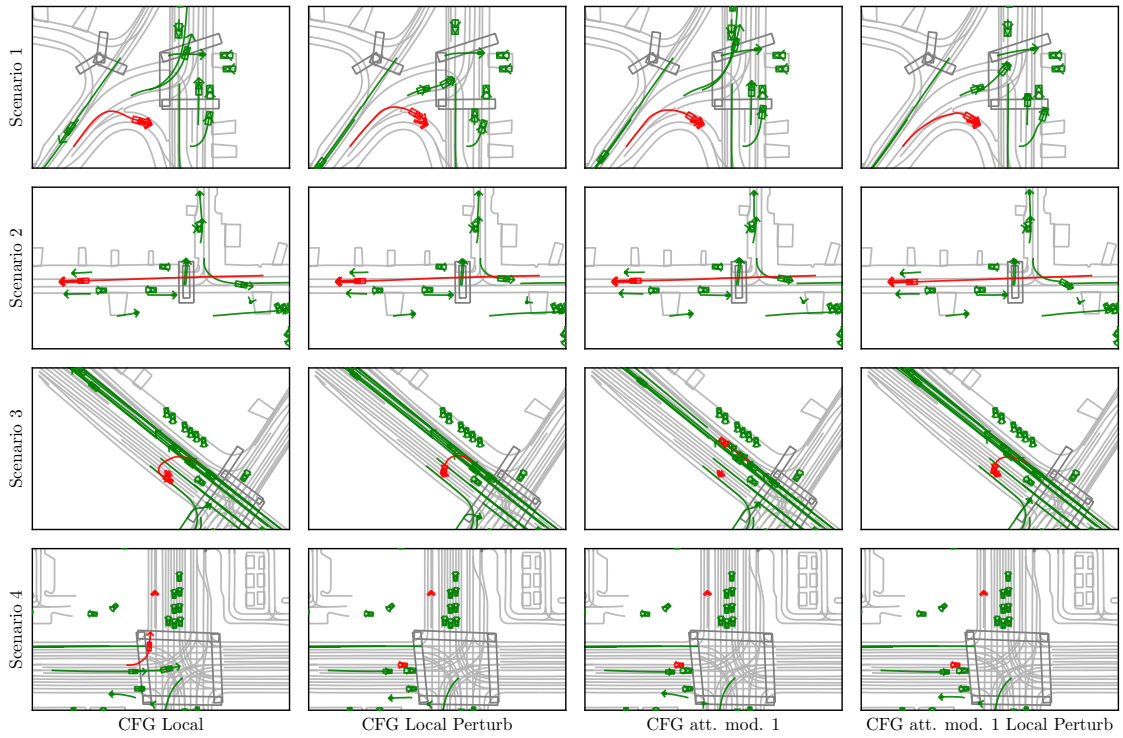


Figure 5.6.2: Scenarios with trajectories generated using the final model and its predecessors with replan 10 and VBD guidance.

For the CFG guidance, the final model performs the best on scenario quality scores. On the target metrics, the final model performs similarly to the CFG Local Perturb model. A worse median, but better mean and deviation shows improved consistency.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	0.906	1.17	1.38	0.0645	0.0846	0.0833	0	0.0301	0.0577	0	0.0141	0.0459
CFG Local Perturb	0.909	1.07	1.25	0.0645	0.0727	0.0767	0	0.0326	0.0586	0	0.0134	0.0448
CFG att. mod. 1	0.890	1.07	1.25	0.0645	0.0780	0.0894	0	0.0328	0.0767	0	0.0122	0.0366
CFG att. mod. 1 Local Perturb	0.864	1.04	1.31	0.0645	0.0719	0.0768	0	0.0332	0.0613	0	$9.32e{-3}$	0.0276

Table 5.6.4: Scenario quality metrics of background agents for the final model and its predecessors with replan 10 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.19	9.91	14.9	9.13	13.4	13.6	2.05	2.67	2.46	0.653	0.947	0.926
CFG Local Perturb	1.09	8.92	13.2	7	11.9	12.3	1.92	2.66	2.47	0.637	0.951	0.909
CFG att. mod. 1	5.63	11.6	15.4	13.5	18.6	18.8	0.914	1.94	3.04	0.395	0.734	0.837
CFG att. mod. 1 Local Perturb	1.67	8.31	11.3	8.17	11.4	10.5	1.72	2.70	2.57	0.575	0.901	0.888

Table 5.6.5: Target metrics for the final model and its predecessors with replan 10 and CFG guidance.

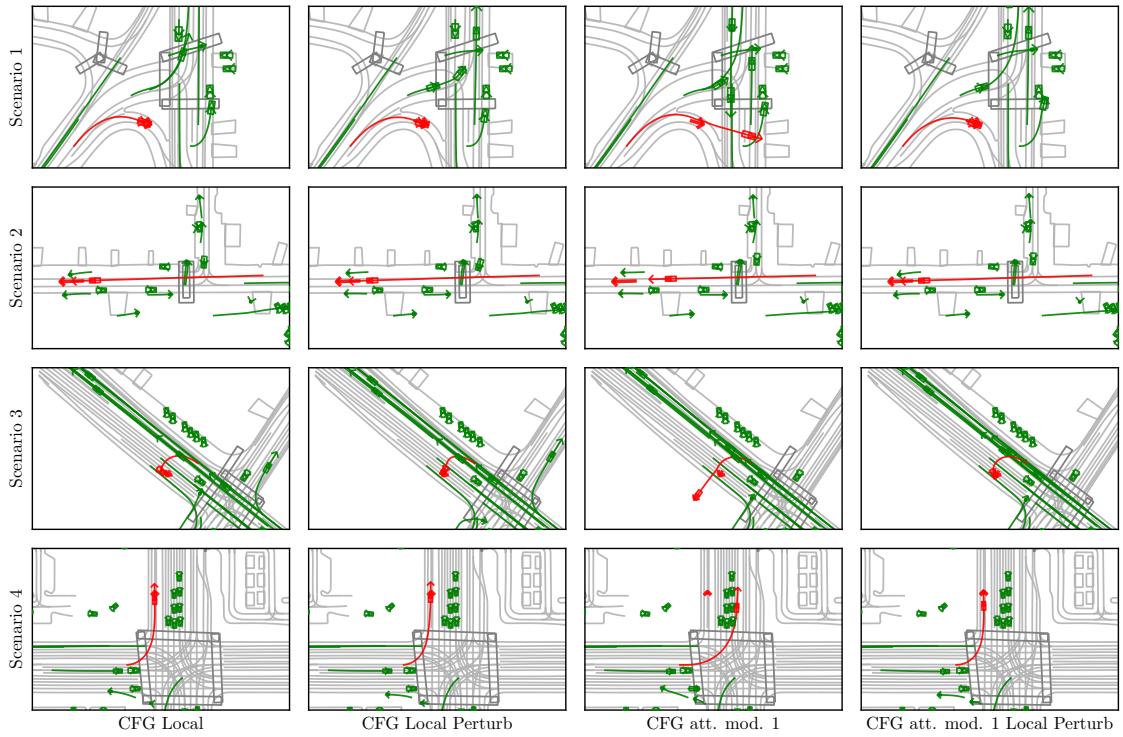


Figure 5.6.3: Scenarios with trajectories generated using the final model and its predecessors with replan 10 and CFG guidance.

The scenario quality scores for the final model using the combined CFG and VBD guidance method is similar to CFG Local Perturb. We observe an improvement in the target metrics scores for the final model, with significant improvements in both the minimum distance and the final distance. while having similar median to the CFG Local model, the mean and std are significantly lower, once again showing the improved consistency.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.03	1.14	1.48	0.0645	0.0890	0.0894	0	0.0297	0.0565	0	0.0140	0.0447
CFG Local Perturb	0.969	1.05	1.36	0.0645	0.0763	0.0807	0	0.0283	0.0413	0	0.0142	0.0458
CFG att. mod. 1	0.889	1.01	1.40	0.0645	0.0806	0.0797	0	0.0327	0.0606	0	0.0115	0.0342
CFG att. mod. 1 Local Perturb	0.899	1.04	1.40	0.0645	0.0749	0.0787	0	0.0326	0.0584	0	$9.64e{-3}$	0.0291

Table 5.6.6: Scenario quality metrics of background agents for the final model and its predecessors with replan 10 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	0.294	1.74	5.07	0.490	2.23	5.13	0.361	1.08	1.83	0.186	0.420	0.626
CFG Local Perturb	0.404	1.82	5.16	0.640	2.33	5.30	0.360	1.06	1.66	0.163	0.459	0.703
CFG att. mod. 1	0.584	2.62	5.97	0.874	3.48	6.37	0.198	0.976	1.66	0.0951	0.422	0.743
CFG att. mod. 1 Local Perturb	0.299	0.877	2.39	0.514	1.49	2.79	0.337	1.12	1.77	0.162	0.469	0.712

Table 5.6.7: Target metrics for the final model and its predecessors with replan 10 and CFG and VBD guidance.

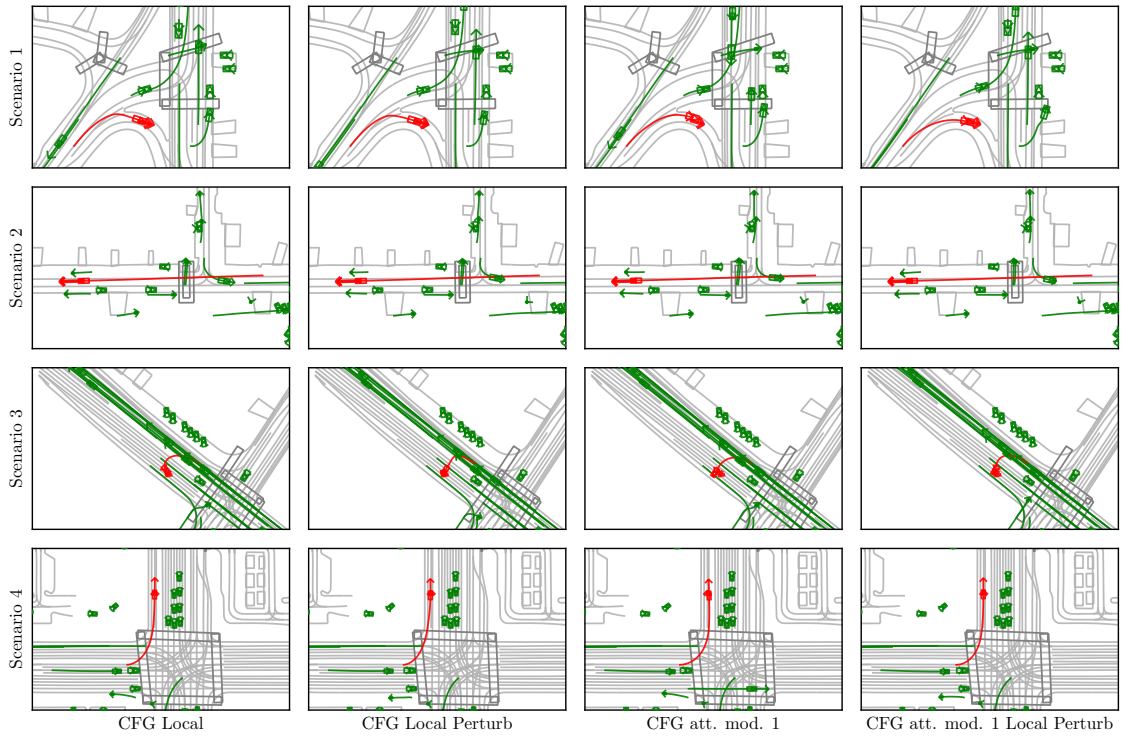


Figure 5.6.4: Scenarios with trajectories generated using the final model and its predecessors with replan 10 and CFG and VBD guidance.

The final model show improvements with respect to all of its predecessors. However, its performance is ultimately a combination between them all, where each metric is comparable to a more specialized model. The greatest improvement seen for the final model is in the distribution of the minimum distance and final distance metrics. Both metrics show a large reduction in the mean and standard deviation, giving the most consistent control over agents out of all its predecessors.

5.7 Guidance Methods

In total there are five different guidance methods that are tested: VBD guidance, with 1 and 10 guidance iteration steps; CFG guidance; and combined CFG and VBD guidance, again, with 1 and 10 guidance iteration steps. Each method have been tested with replan 10 and replan 80. All guidance methods performs better when used with replan 10 compared to replan 80, giving them improved scenario quality metrics scores and target metrics scores. For this reason we will only look at the metrics for trajectories generated with replan 10 in the following section.

To compare the different guidance methods we pick the models with the lowest minimum distance for each guidance type. Table 5.7.1 present the scenario quality scores. They show that scenario quality scores are made worse when the number of guidance iteration steps are increased from one to ten for both VBD guidance and the combined CFG and VBD guidance method. CFG guidance have the best scenario quality scores under the current sorting method.

VBD Guid.		**CFG Guid.**	**Model**	**ADE**			**Collisions**			**Offroad**			**Wrong Way**		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG att. mod. 1	0.999	1.12	1.18	0.0323	0.0578	0.0718	0	0.0282	0.0426	0	0.0106	0.0304
	✓		CFG Local	1.07	1.37	1.60	0.0645	0.0757	0.0846	0	0.0305	0.0581	0	0.0140	0.0445
		✓	CFG att. mod. 1 Local Perturb	0.864	1.04	1.31	0.0645	0.0719	0.0768	0	0.0332	0.0613	0	$9.32e{-3}$	0.0276
✓		✓	CFG att. mod. 1 Local Perturb	0.858	1	1.33	0.0645	0.0747	0.0771	0	0.0304	0.0504	0	$8.82e{-3}$	0.0276
	✓	✓	CFG att. mod. 1 Local Perturb	0.899	1.04	1.40	0.0645	0.0749	0.0787	0	0.0326	0.0584	0	$9.64e{-3}$	0.0291

Table 5.7.1: Scenario quality metrics of background agents for guidance methods with replan 10 and the models with the respective lowest minimum distance.

The target metrics, seen in table 5.7.2, show that each subsequent guidance method improves their target scores compared to the previous method in the table. In contrast to scenario quality, which was impaired by an increase in the number of guidance iteration steps, the target metrics are instead improved.

VBD Guid.		**CFG Guid.**	**Model**	**Minimum Distance**			**Final Distance**			**Speed Error**			**Yaw Error**		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG att. mod. 1	8.24	17.9	23.6	11	19.5	23.5	3.33	4.09	3.37	0.863	1.11	0.931
	✓		CFG Local	1.15	12.8	21.4	3.96	13.7	21.2	1	2.55	3.10	0.521	0.810	0.837
		✓	CFG att. mod. 1 Local Perturb	1.67	8.31	11.3	8.17	11.4	10.5	1.72	2.70	2.57	0.575	0.901	0.888
✓		✓	CFG att. mod. 1 Local Perturb	0.433	1.70	3.94	1.29	2.75	4.25	0.464	1.38	1.99	0.219	0.544	0.751
	✓	✓	CFG att. mod. 1 Local Perturb	0.299	0.877	2.39	0.514	1.49	2.79	0.337	1.12	1.77	0.162	0.469	0.712

Table 5.7.2: Target metrics for guidance methods with replan 10 and the models with the respective lowest minimum distance.

Figure 5.7.1 show trajectories generated with the different guidance methods and models from the previous tables 5.7.1 and 5.7.2. VBD guidance alone with one guidance iteration step gives little apparent control over the agent, while using ten guidance steps improves the control significantly, seen in column one and two. CFG guidance leaves a gap to the target in scenario 2, showing it does not always reach its target, but does come close. The two combined CFG and VBD guidance methods show similar end results, with ten iteration steps giving a slight improvement over using a single iteration, at the cost of a significant increase in additional runtime, as shown previously in section 5.2.

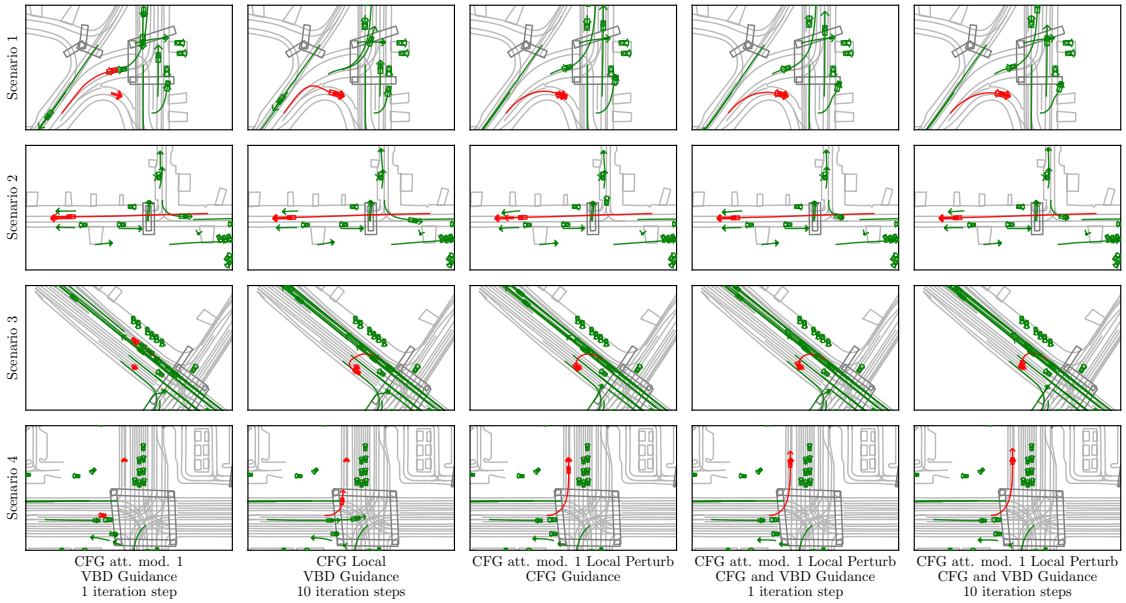


Figure 5.7.1: Scenarios with trajectories generated with replan 10 using the guidance methods and the models with the respective lowest minimum distance.

An important difference between the guidance methods are the values of the median, mean and standard deviation seen for the target metrics. VBD guidance have the largest median, mean and standard deviation of all methods. CFG guidance reduce these values, especially the standard deviation, resulting in more accurate guidance. The combination of CFG and VBD guidance further improve these scores to a significant degree, most notably reducing the mean and standard deviation, making the method much more reliable.

The rank order for the guidance methods, from best to worst, based on the mean minimum distance, are: the combined CFG and VBD guidance, with iter. 10 and iter. 1; CFG guidance; and VBD guidance with iter. 10 and iter. 1.

5.8 Trajectory Optimization

Trajectory optimization is introduced as a fine-tuning step to further improve the target metrics scores for any generated trajectory, independent of the guidance method used. Trajectory optimization is applied after a trajectory has been generated by the model. With replan 80 it is applied once and with replan 10 it is applied eight times. Therefore, the fine-tuning has an advantage with replan 10 and it performs significantly better with it.

The guidance methods follow the same rank order as when trajectory optimization is not used. For the scenario quality metrics, shown in table 5.8.1, we see a lower collisions score but an higher off road score among all methods. The scenario quality metrics are largely unaffected by trajectory optimization.

VBD Guid.		CFG Guid.	Model	ADE			Collisions			Offroad			Wrong Way		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG Local	0.994	1.08	1.30	0.0645	0.0866	0.0874	0	0.0249	0.0380	0	0.0142	0.0453
	✓		CFG Local	1.08	1.30	1.48	0.0645	0.0858	0.0955	0	0.0286	0.0498	0	0.0125	0.0439
		✓	CFG att. mod. 1 Local Perturb	0.814	0.948	1.48	0.0645	0.0864	0.0857	0	0.0280	0.0509	0	$8.93e{-3}$	0.0296
✓		✓	CFG att. mod. 1 Local Perturb	0.901	0.984	1.34	0.0645	0.0817	0.0855	0	0.0285	0.0411	0	$9.80e{-3}$	0.0290
	✓	✓	CFG att. mod. 1 Local Perturb	0.889	1.02	1.42	0.0645	0.0807	0.0883	0	0.0307	0.0508	0	0.0101	0.0308

Table 5.8.1: Scenario quality metrics of background agents for guidance methods with replan 10, trajectory optimization and the models with the respective lowest minimum distance.

For the target metrics, shown in table 5.8.2, trajectory optimization achieves a greatly improved minimum and final distance. The difference between these two metrics is smaller than before, with agents coming closer to their targets. A notable difference is the decreased standard deviation of the minimum distance, which implies that more agents reach their intended destination. Additionally, the minimal distance and final distance are almost equal, which is not the case for the guidance methods when trajectory optimization is not used.

VBD Guid.		CFG Guid.	Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG Local	0.239	8.98	21	0.804	9.39	20.8	1.53	2.93	3.33	0.331	0.841	0.966
	✓		CFG Local	0.164	6.48	18.3	0.324	6.78	18.2	0.332	1.74	2.80	0.125	0.528	0.820
		✓	CFG att. mod. 1 Local Perturb	0.178	0.332	0.590	0.270	0.587	1.12	0.170	0.751	1.47	0.0768	0.304	0.606
✓		✓	CFG att. mod. 1 Local Perturb	0.108	0.166	0.193	0.160	0.324	0.657	0.147	0.873	1.74	0.0648	0.340	0.694
	✓	✓	CFG att. mod. 1 Local Perturb	0.0832	0.127	0.178	0.125	0.360	0.850	0.124	0.761	1.64	0.0608	0.303	0.661

Table 5.8.2: Target metrics for guidance methods with replan 10, trajectory optimization and the models with the respective lowest minimum distance.

In Figure 5.8.1 scenario 1 with CFG guidance shows an example of how the trajectory can be affected by the trajectory optimization step. The trajectory is changed to align with the target, but doing so causes it to take an unnatural path. Scenario 4 with VBD guidance using one iteration step, shows the agent first rotate in place before it is accelerated towards the target, giving an unnatural trajectory. In contrast, the other guided trajectories in scenario 4 show more realistic driving, with the adjustments being spread across the trajectory in time.

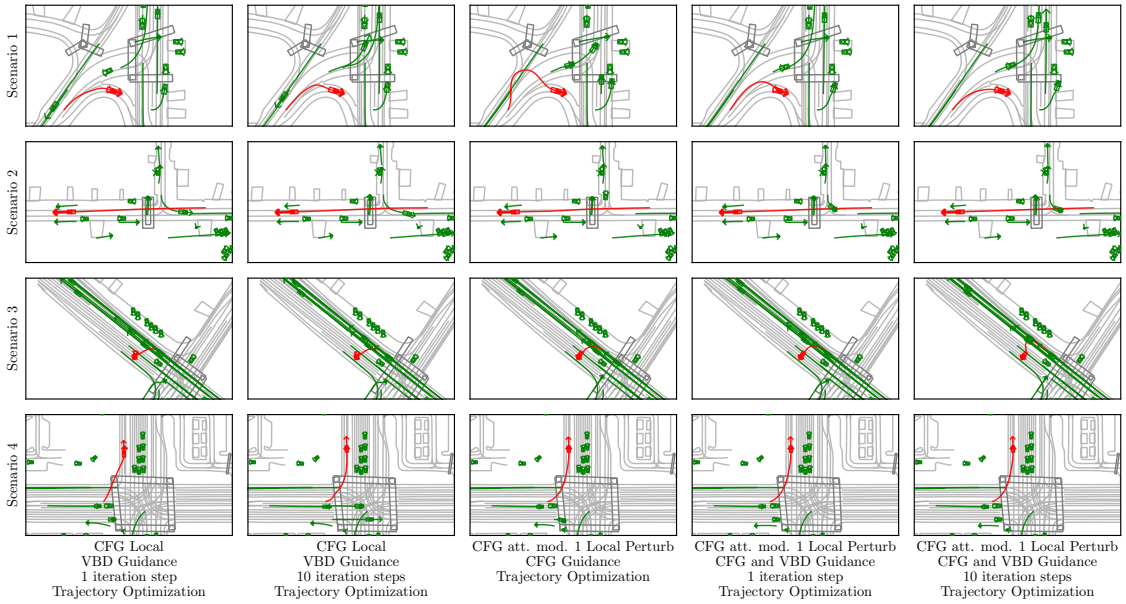


Figure 5.8.1: Scenarios with trajectories generated with replan 10 using the guidance methods with trajectory optimization and the models with the respective lowest minimum distance.

The data from the tables and the examples shown in figure 5.8.1 show that trajectory optimization can significantly improve the trajectories generated, but it also highlights the problems of the method. The optimized trajectory can have uncontrollable turns and twisting trajectories, reducing their quality without going noticed by the current scenario quality metrics. They can also exhibit unrealistic behaviors such as rotating in place, which is a consequence of how the bicycle model is defined. The method would benefit from further constraints to make it more reliable.

6

Conclusion

In this thesis, we implemented and evaluated several models and guidance methods to develop a new approach for steering agent trajectories within arbitrary traffic scenarios. Our experiments produced a diverse set of model-guidance combinations, each exhibiting different strengths and weaknesses in runtime efficiency, scenario fidelity, and controllability.

Building on the observation that prior state-of-the-art methods have inconsistent guidance performance, we hypothesized that their limitations stemmed from a fundamental misalignment between the model prior and the optimization objective. To address this conflict, we utilize a conditioning strategy similar to classifier-free guidance (CFG) in which the model is conditioned on target points during generation. This approach aligns the model prior with the desired optimization behavior. Compared to the existing VBD guidance method, CFG guidance achieves up to a 20× speedup in runtime.

Furthermore, utilizing our final CFG guidance model in combination with VBD guidance yields significantly improved performance. When using the original VBD model and its native guidance method, the system achieves an average final distance of 16.8 m from the desired target points, with a standard deviation of 23.6 m. In contrast, applying our final model with CFG guidance in combination with the existing VBD guidance loop reduces the average final distance to 1.49 m, with the standard deviation reduced to 2.79 m.

In addition to the guidance improvements, we have evaluated a new data augmentation technique called start position perturbation, that improve model robustness by training the model to correct trajectories that deviate from the road markings by moving the start position. Lastly, four different denoiser implementations have been tested to determine an optimal configuration.

Overall, this work supports that aligning model priors with optimization goals, augmenting training data for robustness, and carefully selecting implementation methods can jointly produce a more accurate, efficient, and controllable agent trajectory generation system than previously seen in the literature.

6.1 Answering the Research Questions

The first research question asks how diffusion based motion planning can be guided to fulfill user-specified constraints. We decided to explore constraints through classifier-free guidance, where we condition the model to generate trajectories to reach given targets. These conditioned priors could then be further finetuned using classifier-guidance methods such as VBD guidance for significantly more consistent and accurate trajectory guidance.

The second research question asks how the runtime of the method can be improved to make it appropriate for a test environment for autonomous vehicles. Our choice of using classifier-free guidance allows us to guide the model without needing any additional overhead. Adding a single step of VBD guidance significantly improves performance, while resulting in a significant speedup compared to the original VBD guidance method. Additionally, Since the guidance methods have different guidance performance, quantifying runtime this way might be misleading, since VBD guidance alone often fails to converge to the target.

The third research question asks how the quality of unguided trajectories can be maintained in guided scenarios. We measure this using several metrics: average displacement error, collisions, off road, and wrong way. Our guidance method impacts the quality of background agents negatively. To improve the scenario quality, we introduce a data augmentation technique called start position perturbation. This method improves the scenario quality, but not enough to reach the same level as the original, unguided VBD model.

6.2 Discussion

Our implementation of classifier-free guidance, known as CFG guidance, used for generating trajectories conditioned on targets, achieve satisfactory results. The guidance capability is further improved when it is combined with the original classifier guidance method VBD guidance. The improvement highlights the main limitation of VBD guidance. VBD guidance is based on performing an iterative optimization procedure throughout the denoising process. The number of optimization steps needed to achieve the desired trajectory depends on the similarity of the desired trajectory to the initial one. Since CFG guidance can generate trajectories that are already close to the final trajectory, fewer guidance steps are needed. Additionally, VBD guidance needs to overcome the generative prior of the model to produce the desired trajectory. If it is unable to do this, the model will override the work done by the guidance iteration. This is less of an issue when it is combined with CFG guidance, since the updated trajectory better aligns with the conditioned model, thereby avoiding the misalignment issue with the generative prior. Combining CFG guidance with VBD guidance effectively reduces the computational cost required for producing a trajectory that have successfully converged.

The comparison between relative and local target-encodings show that local targets are more reliable than relative ones. We believe this is due to the increased complex-

ity present in relative targets, making them more difficult to learn. The increased complexity is due to the dependency between the relative target and the orientation of the global scene. Since the orientation is unknown to the model the distance to the target has to be decoded from the relative position, while for local targets the position directly corresponds to the distance and direction of the target. Similarly, the speed needs to be decoded in a similar manner, while local targets represent the speed in the local frame of reference on the agent at the target.

The comparison of the denoiser implementations revealed that relative targets perform similarly to global targets, but that the denoiser implementation does have an impact on performance. All models use global targets, with the possible addition of using relative or local targets in the denoiser. All models except CFG Encoder Only use both target types. In this comparison only relative targets were used, but the models trained with local targets show that the denoiser implementation is working. Models with attention modification 1 and 2 both improved compared to the other models in the test. We have no definitive explanation for the difference in performance. One possibility is that the reduced number of skip connections allows the model to better combine the target embeddings with the intermediate action embeddings as more resources are available. Another possibility is that the target embeddings limit the flexibility of the model, since it is the same embedding used throughout the denoiser. Using separate embedding layers for each skip connection might have resolved the issue.

The evaluation of start position perturbation shows that it improves scenario quality. A side effect of this is a reduced performance when using VBD guidance. We believe this is due to the models improved ability to generate trajectories that adhere to the road markings. The improved driving ability of the model also explains why we see improved performance when using CFG guidance. There is however a discrepancy for the combined guidance method, where CFG is improved, but CFG Local performs worse. An explanation is that the inaccuracy of CFG is improved with the data augmentation, while CFG Local is a bit too eager to reach the target, in spite of not adhering to the road markings when it is possible, i.e. it takes shortcuts. Guidance performance with CFG Local would therefore be reduced if it becomes better at adhering to road markings. These conclusions agree with our observations of individual scenarios.

6.3 Limitations

The present work is subject to several methodological and computational limitations that should be considered when interpreting the results.

First, the simulation scenarios were restricted to a duration of eight seconds. This limits the effect of compounding errors over longer trajectories, therefore hindering our ability to measure the long-horizon performance of the model. As a result, the findings primarily reflect short-term guidance and control performance rather than extended decision-making capabilities.

Second, because of resource constraints, the evaluation of the VBD guidance method was limited to either 10 guidance iterations or a single guidance iteration. Although we have observed anecdotal evidence that the performance in tests conducted with three and six iterations appeared to improve approximately linearly, there are no conclusive results showing this. A more exhaustive evaluation would be required to confirm whether this relationship holds more generally.

Third, the realism metrics employed in the validation process, such as those related to collisions and off-road events, offer only a coarse measure of realism. In their current form, these metrics do not distinguish between minor and severe deviations. For instance, a vehicle that slightly crosses a lane boundary is treated equivalently to one that drives completely off the road. Additionally, the metrics only count whether an agent has experienced an event or not at any time in a scenario. This means that multiple collisions involving the same agent are not differentiated from a single collision. This lack of granularity reduces the interpretability of the realism metrics and may obscure subtle yet important behavioral differences between models.

A further limitation lies in the construction of the validation loop used for guidance evaluation. The current clustering-based target-selection method does not take into account the vehicle's initial velocity. Additionally, while the method only selects targets within road boundaries, it does not ensure that the selected targets lie on the same road as the starting position of the guided vehicle. Consequently, some generated targets may be physically unrealistic or overly difficult to reach. While such challenging scenarios can provide valuable stress-testing for the guidance model, they also complicate the interpretation of performance scores, as an increase in distance to target or off-road cases may stem from the unrealistic nature of certain targets rather than deficiencies in the model itself.

Finally, the available computational resources did not permit a comprehensive hyperparameter search. Parameters such as the learning rate in combination with the target loss, as well as variables related to the guidance process (e.g., the number of guided agents per training scenario), were not systematically optimized. A broader search could potentially yield improved performance or reveal sensitivities in model behavior that remain unexplored in the present study.

6.4 Future Work

Several directions for future research emerge from the limitations and findings of this thesis.

First, a more systematic investigation is warranted into the behavior of agents guided using the Classifier Free guidance approach. The current experiments indicate a notable increase in off-road and collision events when CFG is applied. Future work should therefore aim to determine whether this increase is due to the selection of challenging or unreachable targets during validation, or from limitations inherent to the guidance mechanism itself. One potential avenue for improvement could involve incorporating explicit collision-avoidance or safety constraints within the optimization objective, thereby balancing target adherence with environmental awareness.

In addition, more fine-grained behavior such as lane-change maneuvers, gap acceptance, and yielding behavior merit deeper analysis. Learning these capabilities inherently would improve the models realism and generalizability beyond basic trajectory following.

The issue of CFG overshooting target locations, which is observed consistently, also presents an opportunity for further refinement. Future work could explore enhancements to the temporal embedding in the target information, for instance by improving the representation of target time and integrating it more effectively into the decoder. Such modifications may improve temporal consistency and reduce overshooting tendencies.

From a scalability perspective, future experiments should examine the impact of model size, data quantity, and computational resources. Especially increasing the size of the currently tiny denoiser would be interesting. Larger models trained on broader datasets may demonstrate improved generalization and robustness, but empirical validation is necessary to confirm this hypothesis and to evaluate the associated compute costs.

Finally, an important avenue for future research lies in evaluating the potential of the guidance mechanism for the automatic generation of safety-critical scenarios. If properly utilized, the guidance could serve as a valuable tool for scenario-based testing and validation of autonomous driving systems, particularly in creating high-risk events rare in collected real-world data.

Bibliography

[1] Z. Huang, Z. Zhang, A. Vaidya, Y. Chen, C. Lv, and J. F. Fisac, *Versatile behavior diffusion for generalized traffic agent simulation*, 2024. arXiv: 2404.02524 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2404.02524.

[2] Z. Zhou *et al.*, *Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction*, 2024. arXiv: 2405.17372 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2405.17372.

[3] Y. Wang, T. Zhao, and F. Yi, *Multiverse transformer: 1st place solution for waymo open sim agents challenge 2023*, 2023. arXiv: 2306.11868 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2306.11868.

[4] C. M. Jiang *et al.*, *Scenediffuser: Efficient and controllable driving simulation initialization and rollout*, 2024. arXiv: 2412.12129 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2412.12129.

[5] J. Ho and T. Salimans, *Classifier-free diffusion guidance*, 2022. arXiv: 2207.12598 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2207.12598.

[6] I. J. Goodfellow *et al.*, *Generative adversarial networks*, 2014. arXiv: 1406.2661 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1406.2661.

[7] Y. Xie, X. Guo, C. Wang, K. Liu, and L. Chen, *Advdiffuser: Generating adversarial safety-critical driving scenarios via guided diffusion*, 2024. arXiv: 2410.08453 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2410.08453.

[8] C. Xu, D. Zhao, A. Sangiovanni-Vincentelli, and B. Li, "Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles," in *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=hclEbdHida.

[9] H. Lin *et al.*, *Causal composition diffusion model for closed-loop traffic generation*, 2025. arXiv: 2412.17920 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2412.17920.

[10] P. Dhariwal and A. Nichol, *Diffusion models beat gans on image synthesis*, 2021. arXiv: 2105.05233 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2105.05233.

[11] N. Montali *et al.*, *The waymo open sim agents challenge*, 2023. arXiv: 2305.12032 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2305.12032.

[12] Waymo. "Motion prediction - 2024." (2024), [Online]. Available: https://waymo.com/open/challenges/2024/motion-prediction/ (visited on 08/19/2025).

[13] Waymo. "Sim agents - 2024." (2024), [Online]. Available: https://waymo.com/open/challenges/2024/sim-agents/ (visited on 08/19/2025).

[14] M. Tancik *et al.*, *Fourier features let networks learn high frequency functions in low dimensional domains*, 2020. arXiv: 2006.10739 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2006.10739.

[15] S. Ettinger *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9710–9719.

[16] K. Chen *et al.*, "Womd-lidar: Raw sensor dataset benchmark for motion forecasting," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2024.

[17] H. Caesar *et al.*, "Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *CoRR*, vol. abs/2106.11810, 2021. arXiv: 2106.11810. [Online]. Available: https://arxiv.org/abs/2106.11810.

[18] Z. Zhong *et al.*, *Guided conditional diffusion for controllable traffic simulation*, 2022. arXiv: 2210.17366 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2210.17366.

[19] J. Houston *et al.*, "One thousand and one hours: Self-driving motion prediction dataset," *CoRR*, vol. abs/2006.14480, 2020. arXiv: 2006.14480. [Online]. Available: https://arxiv.org/abs/2006.14480.

[20] S. Shi, L. Jiang, D. Dai, and B. Schiele, *Motion transformer with global intention localization and local movement refinement*, 2023. arXiv: 2209.13508 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2209.13508.

[21] Neelofar and A. Aleti, *Identifying and explaining safety-critical scenarios for autonomous vehicles via key features*, 2023. arXiv: 2212.07566 [cs.SE]. [Online]. Available: https://arxiv.org/abs/2212.07566.

[22] F. Hauer, I. Gerostathopoulos, T. Schmidt, and A. Pretschner, "Clustering traffic scenarios using mental models as little as possible," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1007–1012. DOI: 10.1109/IV47402.2020.9304636.

[23] X. Zhang *et al.*, "Finding critical scenarios for automated driving systems: A systematic mapping study," *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 991–1026, 2023. DOI: 10.1109/TSE.2022.3170122.

[24] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, *Generating useful accident-prone driving scenarios via a learned traffic prior*, 2022. arXiv: 2112.05077 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2112.05077.

[25] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, *King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients*, 2022. arXiv: 2204.13683 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2204.13683.

[26] Y. Yin, P. Khayatan, É. Zablocki, A. Boulch, and M. Cord, *Regents: Real-world safety-critical driving scenario generation made stable*, 2024. arXiv: 2409.07830 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2409.07830.

[27] Z. Wang *et al.*, "Safety evaluation of autonomous driving based on safety-critical scenario generation," in *Intelligent Transportation Systems (ITSC)*, Edmonton, Canada, Sep. 2024.

[28] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, 2020. arXiv: 2006.11239 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2006.11239.

[29] W. Ding, B. Chen, B. Li, K. J. Eun, and D. Zhao, *Multimodal safety-critical scenarios generation for decision-making algorithms evaluation*, 2020. arXiv: 2009.08311 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2009.08311.

A

Appendix 1

For completeness we include the results of all methods when used with replan 80. The results follow the same order as in the results section.

A.1 Relative and Local Targets

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.918	1.12	0.928	0.0323	0.0700	0.0842	0	0.0272	0.0678	0	$8.31e{-3}$	0.0400
CFG	0.949	1.24	1.19	0.0645	0.0813	0.0848	0	0.0316	0.0818	0	$9.69e{-3}$	0.0394
CFG Local	0.956	1.25	1.18	0.0645	0.0714	0.0821	0	0.0335	0.0834	0	0.0100	0.0397

Table A.1.1: Scenario quality metrics of background agents for relative target and local target models with replan 80 and no guidance.

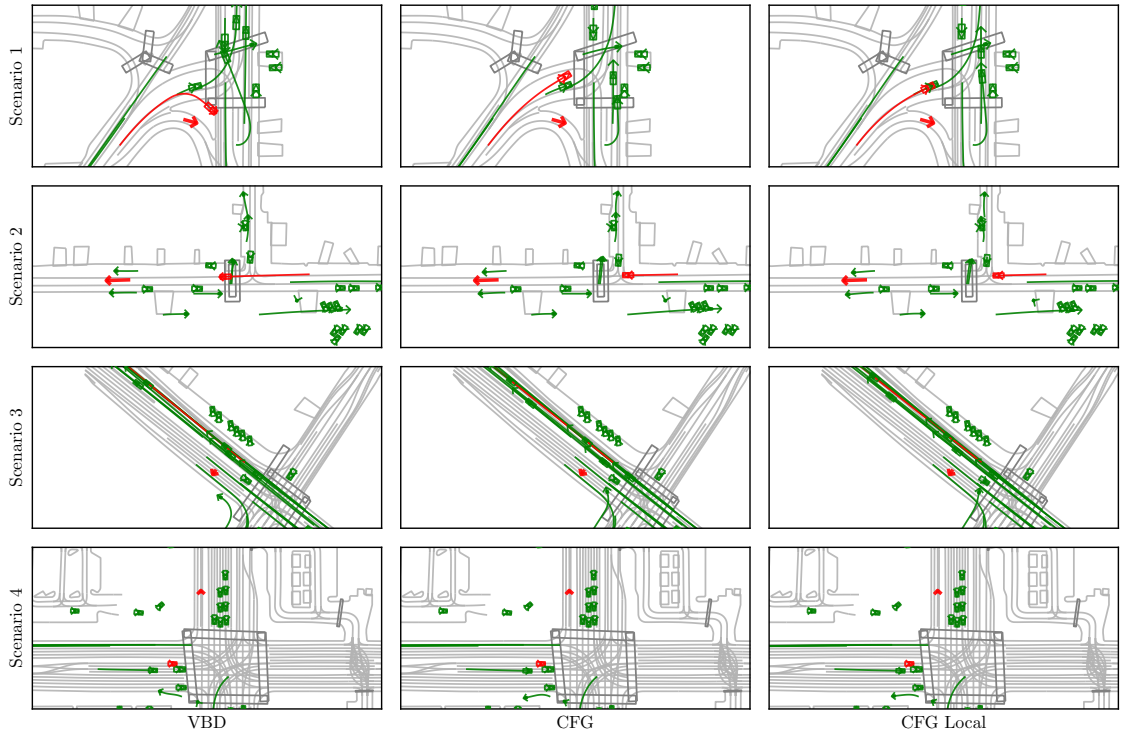


Figure A.1.1: Scenarios with trajectories generated using relative target and local target models with replan 80 and no guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	1.20	1.74	2.33	0.0968	0.108	0.106	0	0.0350	0.0658	0	$8.71e{-3}$	0.0252
CFG	1.28	1.68	1.82	0.0968	0.112	0.110	0	0.0367	0.0655	0	$7.44e{-3}$	0.0207
CFG Local	1.24	1.64	1.67	0.0938	0.111	0.108	0	0.0384	0.0709	0	0.0131	0.0418

Table A.1.2: Scenario quality metrics of background agents for relative target and local target models with replan 80 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	7.25	17.9	24.3	8.89	18.9	24	4.26	4.72	3.50	1.12	1.25	0.944
CFG	5.53	17.7	24.2	7.51	18.5	24	3.83	4.54	3.61	0.992	1.21	0.954
CFG Local	4.31	16	22.3	5.31	16.9	22	3.76	4.38	3.44	1	1.20	0.968

Table A.1.3: Target metrics for relative target and local target models with replan 80 and VBD guidance.

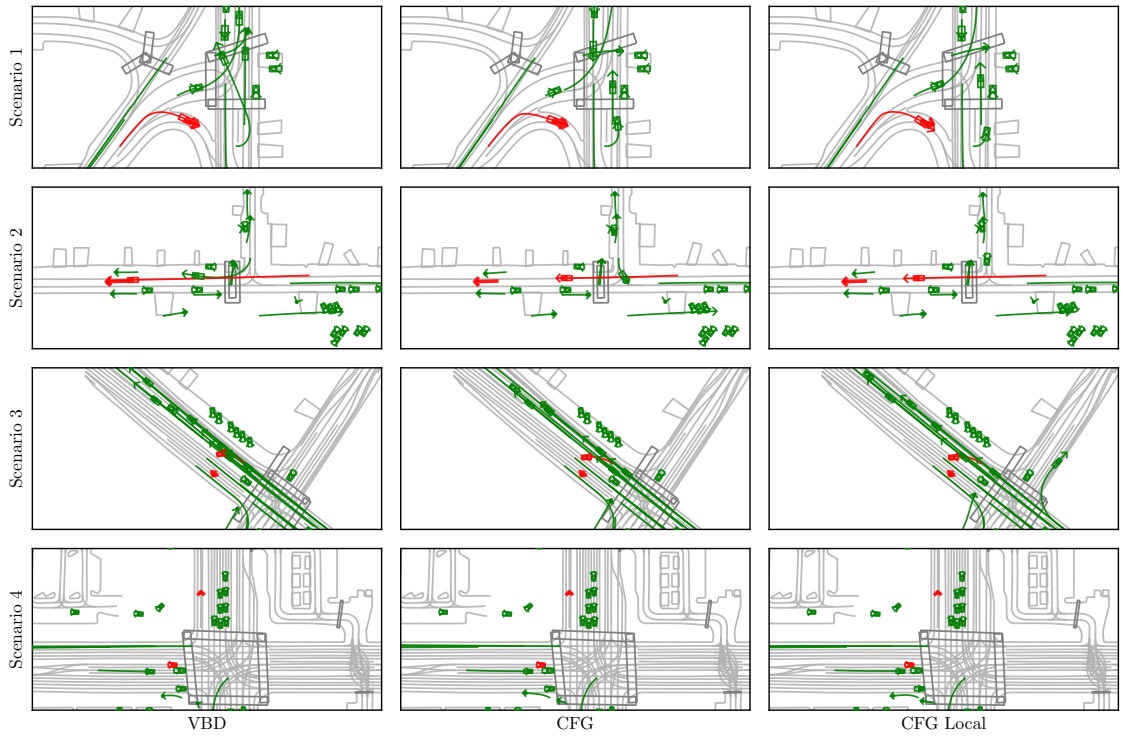


Figure A.1.2: Scenarios with trajectories generated using relative target and local target models with replan 80 and VBD guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	1.06	1.30	1.52	0.0968	0.106	0.0924	0	0.0346	0.0835	0	0.0112	0.0398
CFG Local	1.05	1.29	1.57	0.0968	0.103	0.0916	0	0.0340	0.0644	0	0.0114	0.0411

Table A.1.4: Scenario quality metrics of background agents for relative target and local target models with replan 80 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	3.55	8.34	11.5	9.29	13	12.8	1.18	1.94	2.12	0.521	0.842	0.861
CFG Local	1.17	5.33	8.60	8.24	11.3	11.1	0.975	1.60	1.67	0.442	0.754	0.798

Table A.1.5: Target metrics for relative target and local target models with replan 80 and CFG guidance.

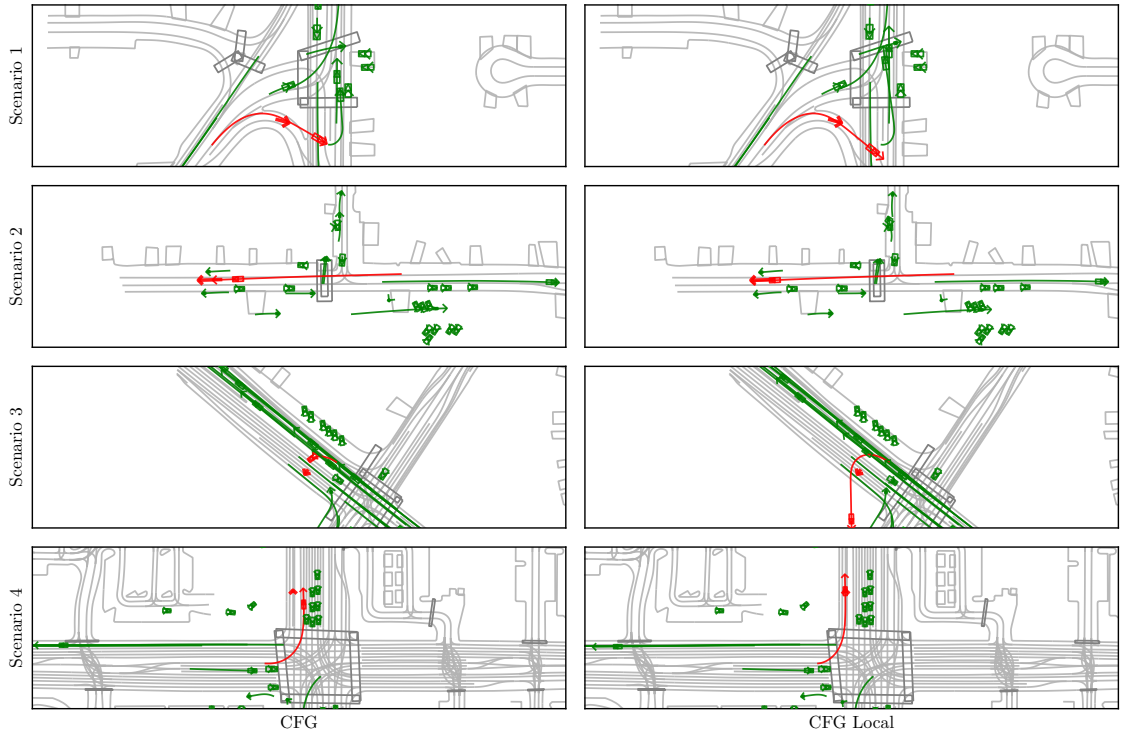


Figure A.1.3: Scenarios with trajectories generated using relative target and local target models with replan 80 and CFG guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	1.03	1.29	1.67	0.0968	0.114	0.0986	0	0.0278	0.0580	0	$7.05e{-3}$	0.0234
CFG Local	1.02	1.29	1.82	0.0968	0.109	0.0986	0	0.0342	0.0837	0	0.0107	0.0264

Table A.1.6: Scenario quality metrics of background agents for relative target and local target models with replan 80 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.657	1.97	3.84	1.14	2.48	4.42	0.691	1.60	1.92	0.337	0.668	0.790
CFG Local	0.539	1.46	3.03	0.821	1.89	3.21	0.565	1.56	1.98	0.266	0.602	0.778

Table A.1.7: Target metrics for relative target and local target models with replan 80 and CFG and VBD guidance.

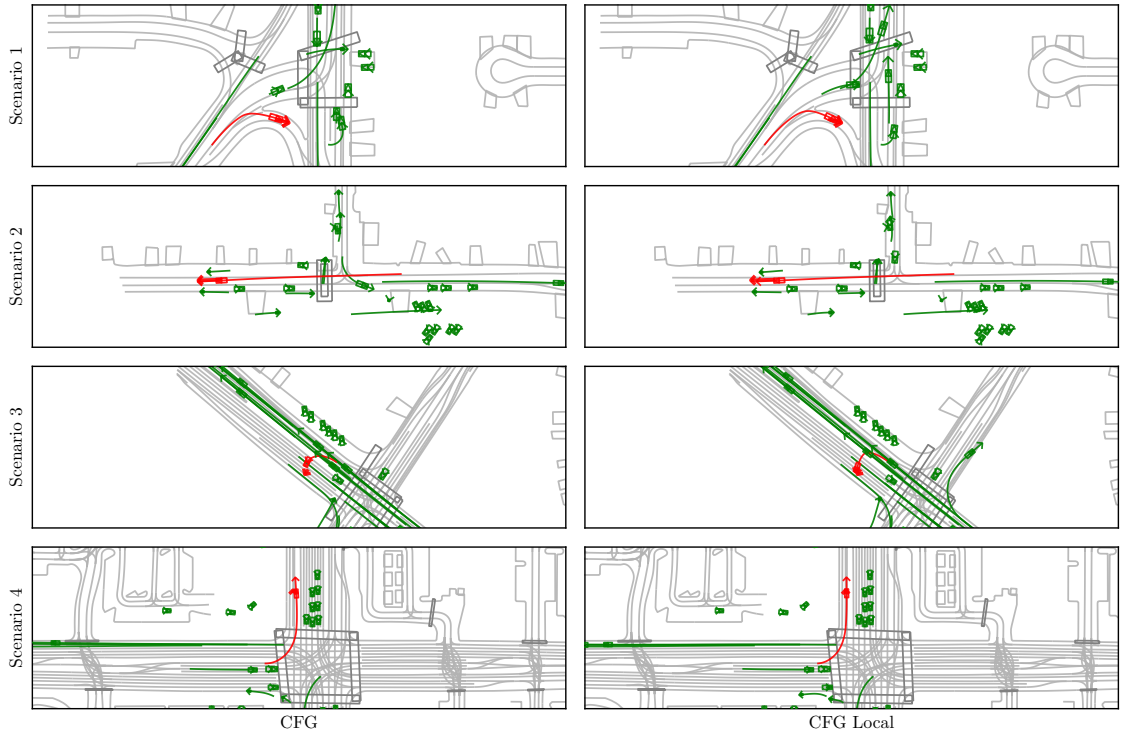


Figure A.1.4: Scenarios with trajectories generated using relative target and local target models with replan 80 and CFG and VBD guidance.

A.2 Start Position Perturbation

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.918	1.12	0.928	0.0323	0.0700	0.0842	0	0.0272	0.0678	0	$8.31e{-3}$	0.0400
VBD Perturb	0.897	1.07	0.860	0.0323	0.0679	0.0815	0	0.0200	0.0401	0	$9.58e{-3}$	0.0393
CFG	0.949	1.24	1.19	0.0645	0.0813	0.0848	0	0.0316	0.0818	0	$9.69e{-3}$	0.0394
CFG Perturb	0.898	1.10	0.992	0.0645	0.0759	0.0845	0	0.0341	0.0726	0	$9.21e{-3}$	0.0391
CFG Local	0.956	1.25	1.18	0.0645	0.0714	0.0821	0	0.0335	0.0834	0	0.0100	0.0397
CFG Local Perturb	0.968	1.18	1.08	0.0645	0.0738	0.0809	0	0.0352	0.0823	0	$8.40e{-3}$	0.0389

Table A.2.1: Scenario quality metrics of background agents for baseline models and data augmented models with replan 80 and no guidance.

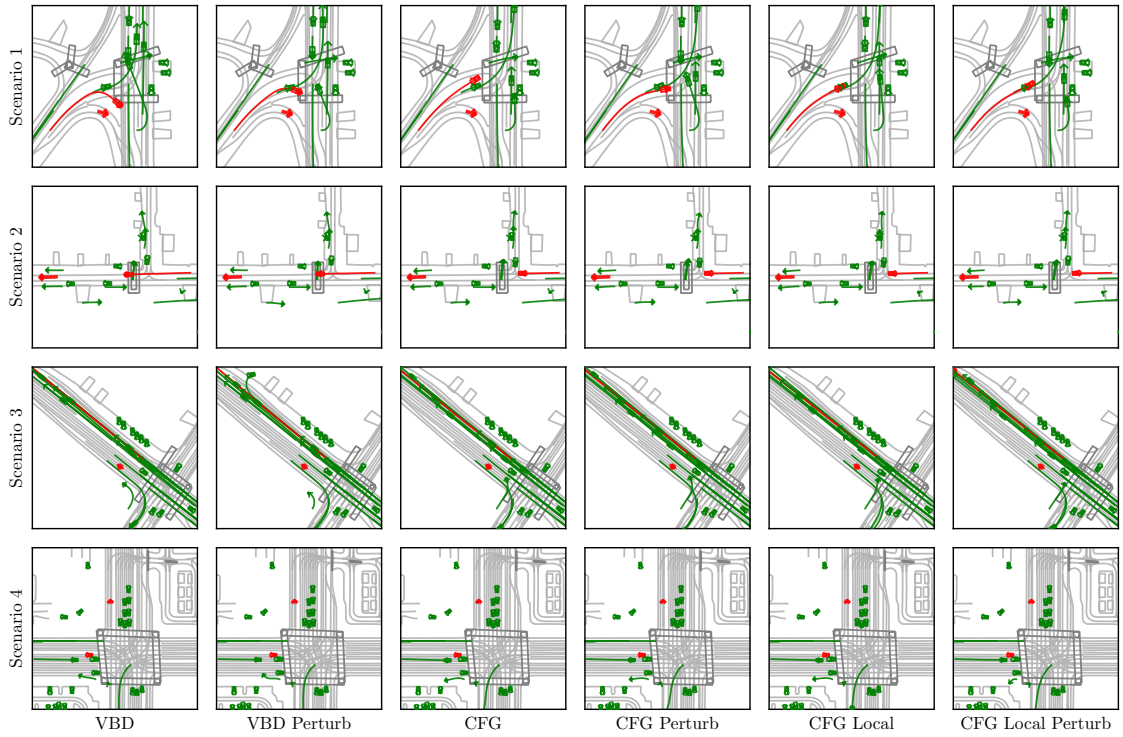


Figure A.2.1: Scenarios with trajectories generated using baseline models and data augmented models with replan 80 and no guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	1.20	1.74	2.33	0.0968	0.108	0.106	0	0.0350	0.0658	0	$8.71e{-3}$	0.0252
VBD Perturb	1.08	1.62	2.06	0.0714	0.104	0.113	0	0.0330	0.0673	0	$9.18e{-3}$	0.0388
CFG	1.28	1.68	1.82	0.0968	0.112	0.110	0	0.0367	0.0655	0	$7.44e{-3}$	0.0207
CFG Perturb	1.10	1.44	1.62	0.0968	0.111	0.107	0	0.0435	0.0897	0	0.0123	0.0325
CFG Local	1.24	1.64	1.67	0.0938	0.111	0.108	0	0.0384	0.0709	0	0.0131	0.0418
CFG Local Perturb	1.11	1.63	2.20	0.0931	0.108	0.113	0.0318	0.0375	0.0634	0	0.0101	0.0341

Table A.2.2: Scenario quality metrics of background agents for baseline models and data augmented models with replan 80 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	7.25	17.9	24.3	8.89	18.9	24	4.26	4.72	3.50	1.12	1.25	0.944
VBD Perturb	4.69	17.3	24.2	5.95	18	23.9	3.81	4.57	3.60	1	1.23	0.987
CFG	5.53	17.7	24.2	7.51	18.5	24	3.83	4.54	3.61	0.992	1.21	0.954
CFG Perturb	7.20	17.7	24.6	8.40	18.5	24.3	3.83	4.51	3.46	1.02	1.22	0.958
CFG Local	4.31	16	22.3	5.31	16.9	22	3.76	4.38	3.44	1	1.20	0.968
CFG Local Perturb	6.05	18.6	25.1	6.99	19.4	25	3.98	4.71	3.50	1.15	1.28	0.989

Table A.2.3: Target metrics for baseline models and data augmented models with replan 80 and VBD guidance.

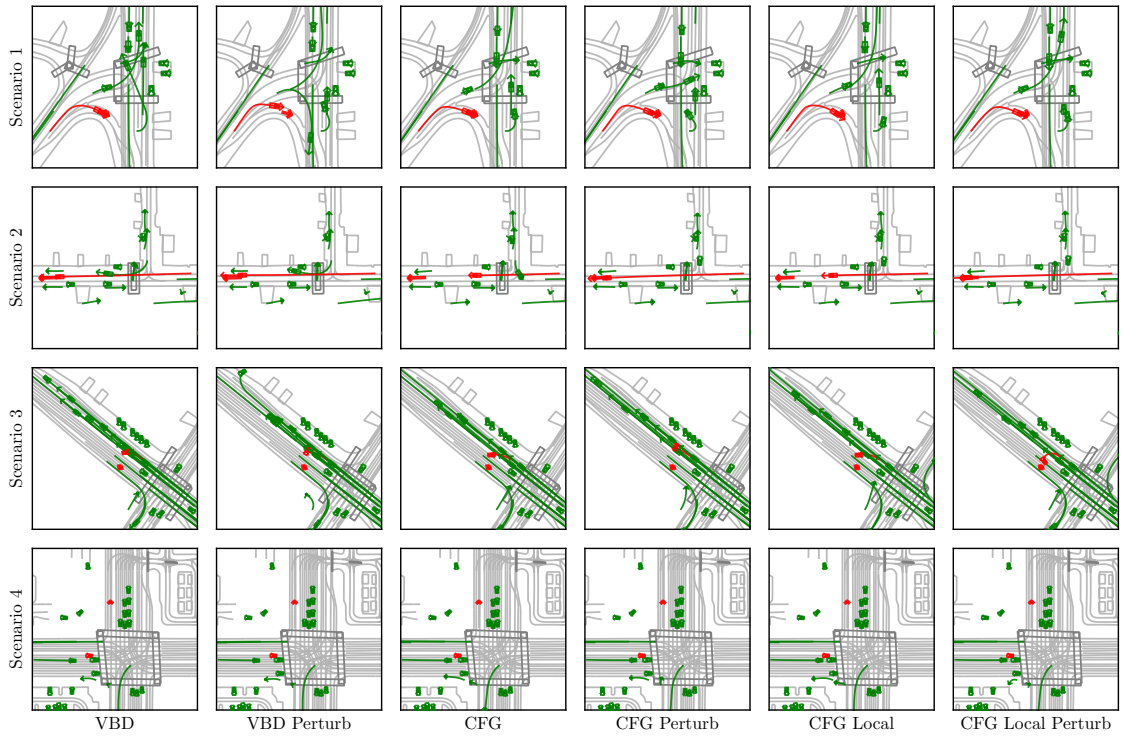


Figure A.2.2: Scenarios with trajectories generated using baseline models and data augmented models with replan 80 and VBD guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	1.06	1.30	1.52	0.0968	0.106	0.0924	0	0.0346	0.0835	0	0.0112	0.0398
CFG Perturb	0.925	1.16	1.49	0.0968	0.108	0.0975	0	0.0364	0.0736	0	$8.77e{-3}$	0.0385
CFG Local	1.05	1.29	1.57	0.0968	0.103	0.0916	0	0.0340	0.0644	0	0.0114	0.0411
CFG Local Perturb	1.03	1.19	1.50	0.0968	0.104	0.0914	0	0.0360	0.0831	0	$8.98e{-3}$	0.0386

Table A.2.4: Scenario quality metrics of background agents for baseline models and data augmented models with replan 80 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	3.55	8.34	11.5	9.29	13	12.8	1.18	1.94	2.12	0.521	0.842	0.861
CFG Perturb	2.53	6.66	9.58	9	11.2	9.81	1.18	1.99	2.10	0.557	0.850	0.858
CFG Local	1.17	5.33	8.60	8.24	11.3	11.1	0.975	1.60	1.67	0.442	0.754	0.798
CFG Local Perturb	0.858	4.30	7.74	6.75	9.73	9.99	1.24	1.91	1.91	0.596	0.844	0.829

Table A.2.5: Target metrics for baseline models and data augmented models with replan 80 and CFG guidance.

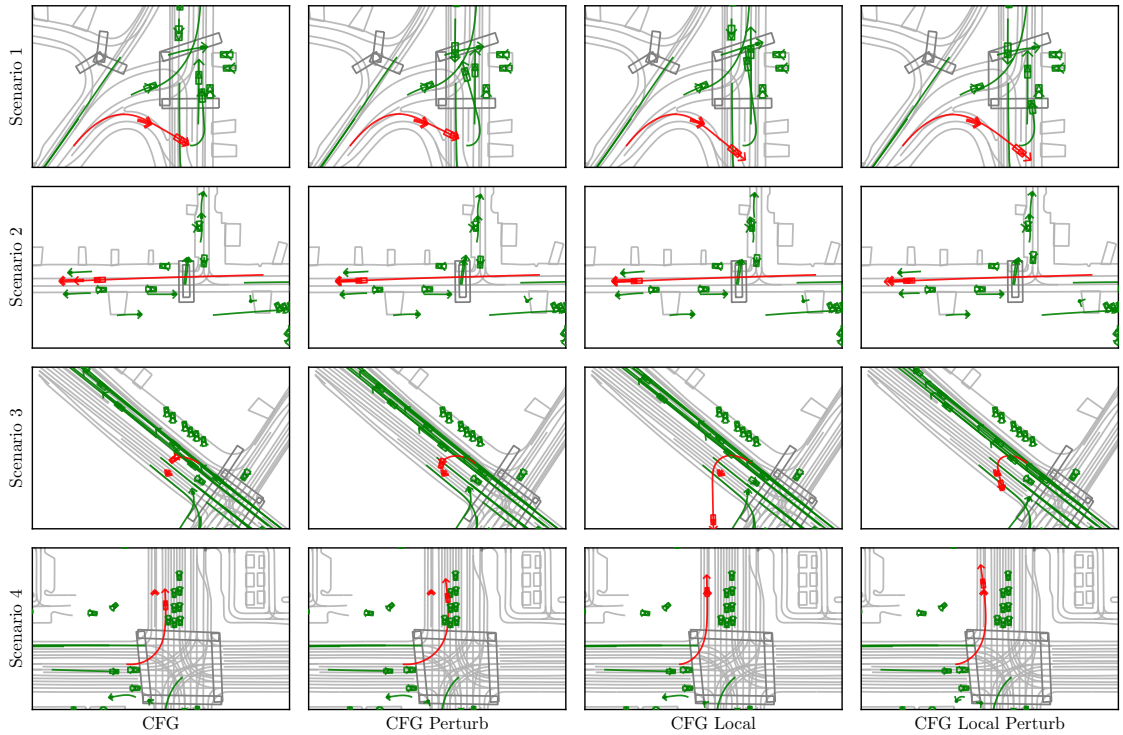


Figure A.2.3: Scenarios with trajectories generated using baseline models and data augmented models with replan 80 and CFG guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	1.03	1.29	1.67	0.0968	0.114	0.0986	0	0.0278	0.0580	0	$7.05e{-3}$	0.0234
CFG Perturb	0.927	1.14	1.46	0.0851	0.103	0.0979	0	0.0353	0.0659	0	$9.34e{-3}$	0.0283
CFG Local	1.02	1.29	1.82	0.0968	0.109	0.0986	0	0.0342	0.0837	0	0.0107	0.0264
CFG Local Perturb	0.976	1.20	1.45	0.0968	0.105	0.0908	0	0.0361	0.0701	0	0.0101	0.0396

Table A.2.6: Scenario quality metrics of background agents for baseline models and data augmented models with replan 80 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.657	1.97	3.84	1.14	2.48	4.42	0.691	1.60	1.92	0.337	0.668	0.790
CFG Perturb	0.631	1.26	1.90	1.15	1.82	2.25	0.672	1.58	2.02	0.307	0.690	0.827
CFG Local	0.539	1.46	3.03	0.821	1.89	3.21	0.565	1.56	1.98	0.266	0.602	0.778
CFG Local Perturb	0.593	1.97	4.97	0.954	2.38	4.96	0.724	1.58	1.98	0.326	0.645	0.775

Table A.2.7: Target metrics for baseline models and data augmented models with replan 80 and CFG and VBD guidance.

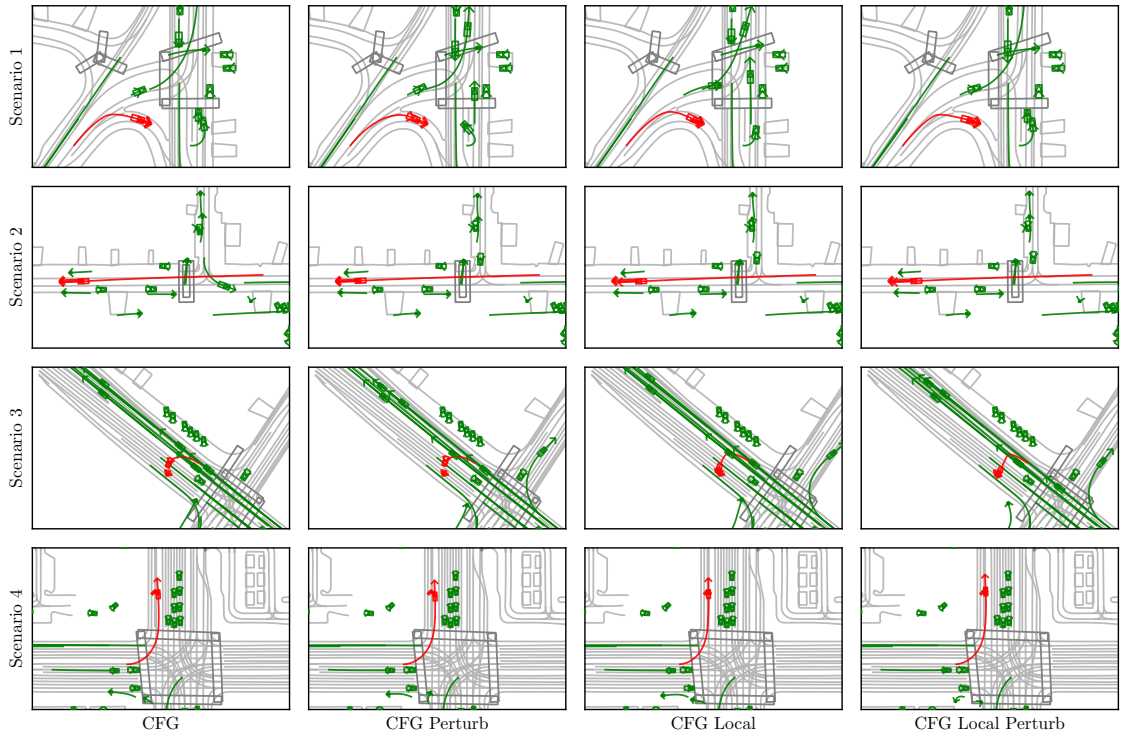


Figure A.2.4: Scenarios with trajectories generated using baseline models and data augmented models with replan 80 and CFG and VBD guidance.

A.3 Denoiser Implementations

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	0.918	1.12	0.928	0.0323	0.0700	0.0842	0	0.0272	0.0678	0	$8.31e{-3}$	0.0400
CFG	0.949	1.24	1.19	0.0645	0.0813	0.0848	0	0.0316	0.0818	0	$9.69e{-3}$	0.0394
CFG att. mod. 1	0.918	1.14	1.03	0.0645	0.0777	0.0886	0	0.0293	0.0751	0	$7.75e{-3}$	0.0271
CFG att. mod. 2	0.896	1.19	1.08	0.0617	0.0748	0.0905	0	0.0309	0.0745	0	0.0119	0.0409
CFG Enc. only	0.956	1.26	1.23	0.0645	0.0814	0.0853	0	0.0327	0.0817	0	$9.29e{-3}$	0.0385

Table A.3.1: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 80 and no guidance.

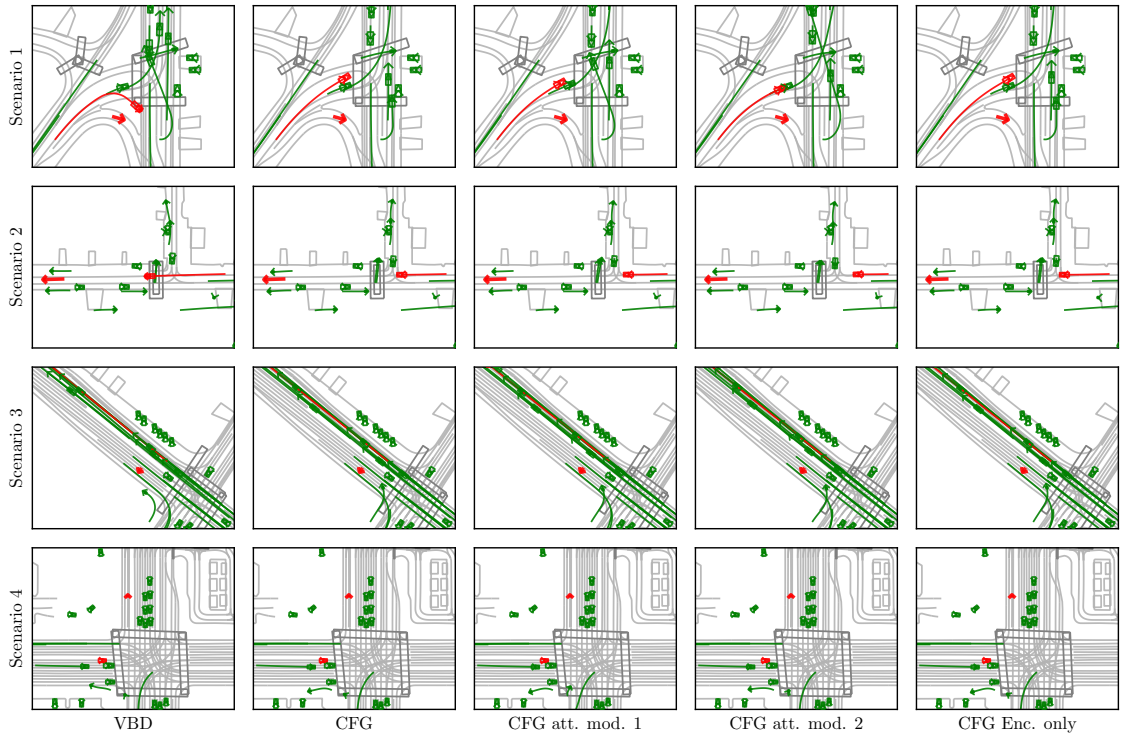


Figure A.3.1: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 80 and no guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	1.20	1.74	2.33	0.0968	0.108	0.106	0	0.0350	0.0658	0	$8.71e{-3}$	0.0252
CFG	1.28	1.68	1.82	0.0968	0.112	0.110	0	0.0367	0.0655	0	$7.44e{-3}$	0.0207
CFG att. mod. 1	1.21	1.65	1.80	0.0968	0.104	0.0972	0	0.0365	0.0598	0	$8.38e{-3}$	0.0255
CFG att. mod. 2	1.27	1.75	1.86	0.0909	0.105	0.105	0	0.0381	0.0618	0	0.0131	0.0348
CFG Enc. only	1.27	1.65	1.73	0.0968	0.117	0.116	0	0.0346	0.0642	0	$7.31e{-3}$	0.0209

Table A.3.2: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 80 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
VBD	7.25	17.9	24.3	8.89	18.9	24	4.26	4.72	3.50	1.12	1.25	0.944
CFG	5.53	17.7	24.2	7.51	18.5	24	3.83	4.54	3.61	0.992	1.21	0.954
CFG att. mod. 1	3.90	16.9	23.3	5.91	17.9	23.1	3.75	4.37	3.53	0.990	1.18	0.963
CFG att. mod. 2	5.38	18	24	7.37	19	23.8	3.99	4.59	3.47	1.04	1.21	0.957
CFG Enc. only	4.41	17.8	24.5	7.52	18.6	24.3	3.77	4.53	3.61	1	1.17	0.950

Table A.3.3: Target metrics for the four models using different denoiser implementations with replan 80 and VBD guidance.

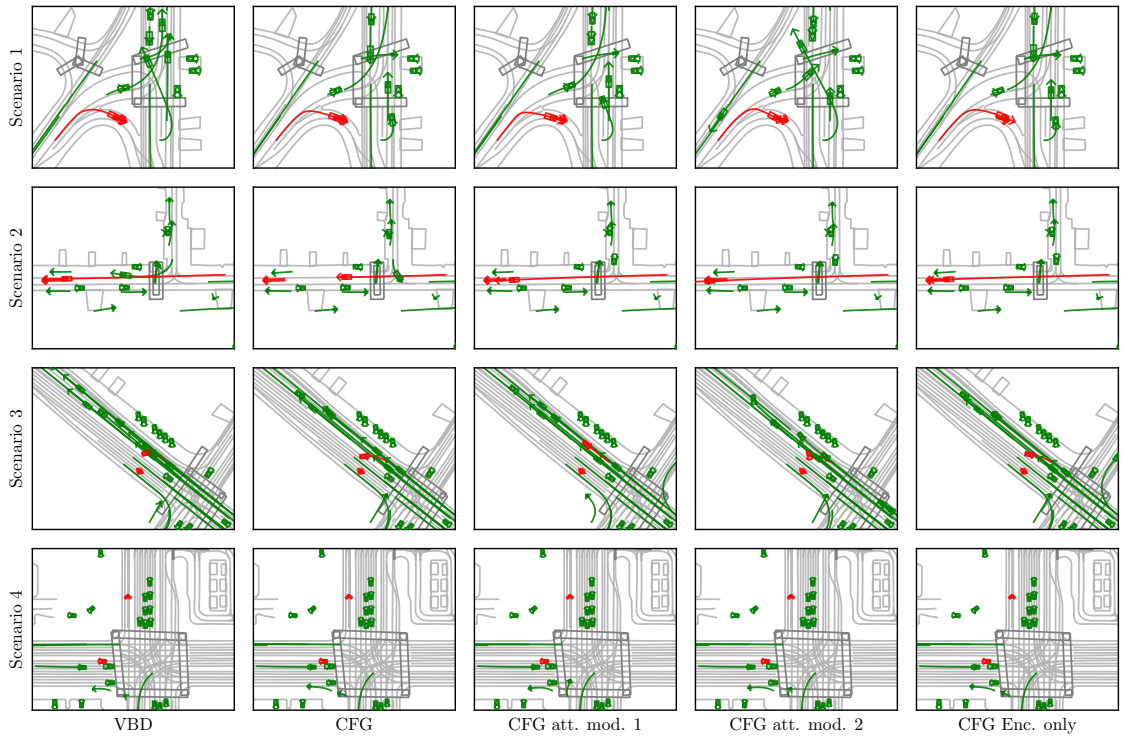


Figure A.3.2: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 80 and VBD guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	1.06	1.30	1.52	0.0968	0.106	0.0924	0	0.0346	0.0835	0	0.0112	0.0398
CFG att. mod. 1	0.980	1.18	1.49	0.0968	0.108	0.0974	0	0.0307	0.0724	0	$7.81e{-3}$	0.0270
CFG att. mod. 2	0.930	1.21	1.57	0.0968	0.105	0.0950	0	0.0369	0.0816	0	0.0131	0.0417
CFG Enc. only	1.05	1.30	1.57	0.0968	0.109	0.0947	0	0.0357	0.0849	0	$9.97e{-3}$	0.0388

Table A.3.4: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 80 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	3.55	8.34	11.5	9.29	13	12.8	1.18	1.94	2.12	0.521	0.842	0.861
CFG att. mod. 1	2.63	6.73	9.86	8.55	11.8	12.3	1.21	1.91	2	0.578	0.872	0.864
CFG att. mod. 2	2.38	6.74	10.1	8.67	11.6	11.2	1.20	1.85	1.98	0.487	0.809	0.823
CFG Enc. only	3.54	8.30	11.5	9.33	13	13	1.24	1.95	2.15	0.522	0.827	0.846

Table A.3.5: Target metrics for the four models using different denoiser implementations with replan 80 and CFG guidance.

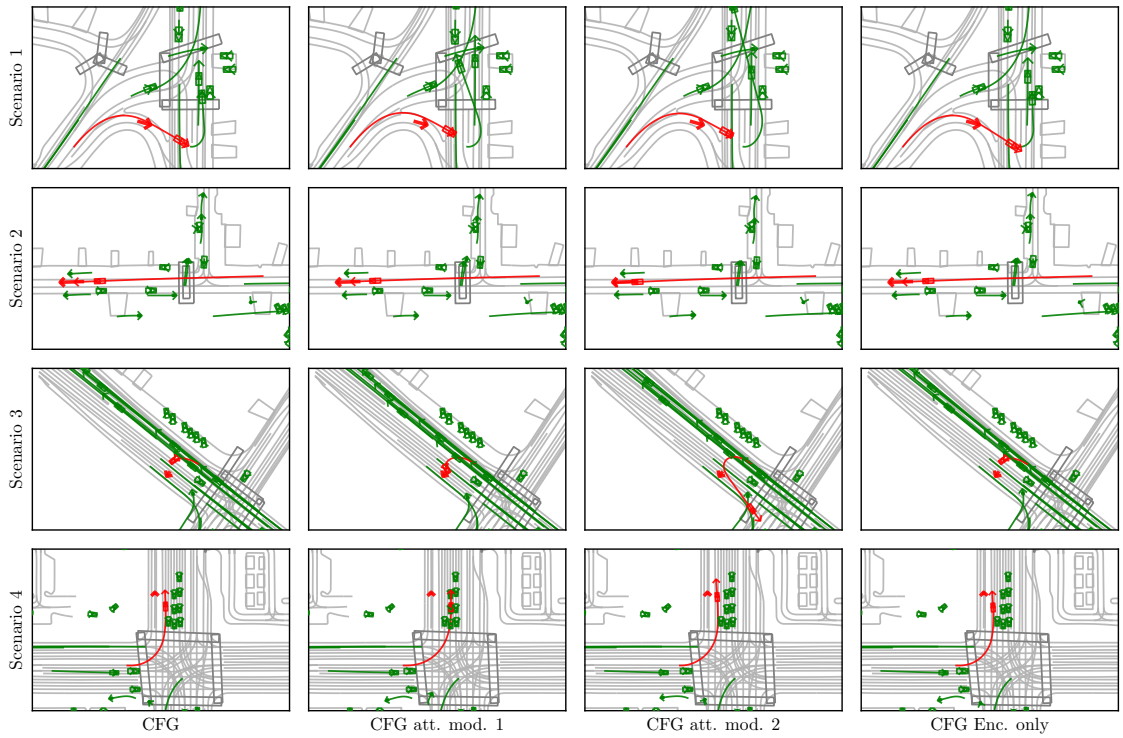


Figure A.3.3: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 80 and CFG guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	1.03	1.29	1.67	0.0968	0.114	0.0986	0	0.0278	0.0580	0	$7.05e{-3}$	0.0234
CFG att. mod. 1	0.995	1.14	1.60	0.0938	0.103	0.0931	0	0.0297	0.0556	0	$8.71e{-3}$	0.0242
CFG att. mod. 2	0.974	1.23	1.75	0.0968	0.112	0.0964	0	0.0295	0.0481	0	$9.20e{-3}$	0.0252
CFG Enc. only	1.03	1.28	1.57	0.0968	0.116	0.0967	0	0.0302	0.0587	0	$7.52e{-3}$	0.0236

Table A.3.6: Scenario quality metrics of background agents for the four models using different denoiser implementations with replan 80 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG	0.657	1.97	3.84	1.14	2.48	4.42	0.691	1.60	1.92	0.337	0.668	0.790
CFG att. mod. 1	0.657	1.72	3.34	1.05	2.24	4.05	0.611	1.48	1.89	0.268	0.616	0.768
CFG att. mod. 2	0.740	1.59	2.73	1.14	2.08	2.88	0.740	1.62	1.93	0.308	0.682	0.821
CFG Enc. only	0.688	2.03	3.98	1.19	2.49	4.49	0.693	1.66	1.94	0.326	0.700	0.818

Table A.3.7: Target metrics for the four models using different denoiser implementations with replan 80 and CFG and VBD guidance.

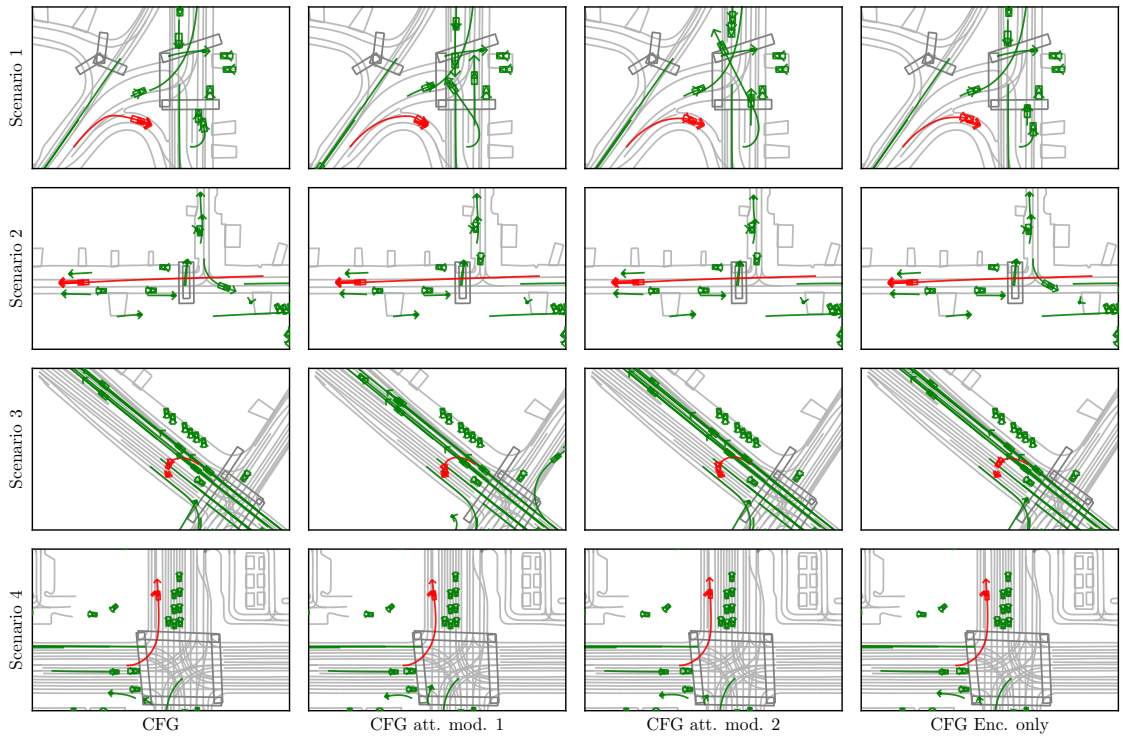


Figure A.3.4: Scenarios with trajectories generated using the four models using different denoiser implementations with replan 80 and CFG and VBD guidance.

A.4 Final Model

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	0.956	1.25	1.18	0.0645	0.0714	0.0821	0	0.0335	0.0834	0	0.0100	0.0397
CFG Local Perturb	0.968	1.18	1.08	0.0645	0.0738	0.0809	0	0.0352	0.0823	0	$8.40e{-3}$	0.0389
CFG att. mod. 1	0.918	1.14	1.03	0.0645	0.0777	0.0886	0	0.0293	0.0751	0	$7.75e{-3}$	0.0271
CFG att. mod. 1 Local Perturb	0.884	1.15	1.14	0.0645	0.0789	0.0864	0	0.0361	0.0904	0	0.0105	0.0392

Table A.4.1: Scenario quality metrics of background agents for the final model and its predecessors with replan 80 and no guidance.

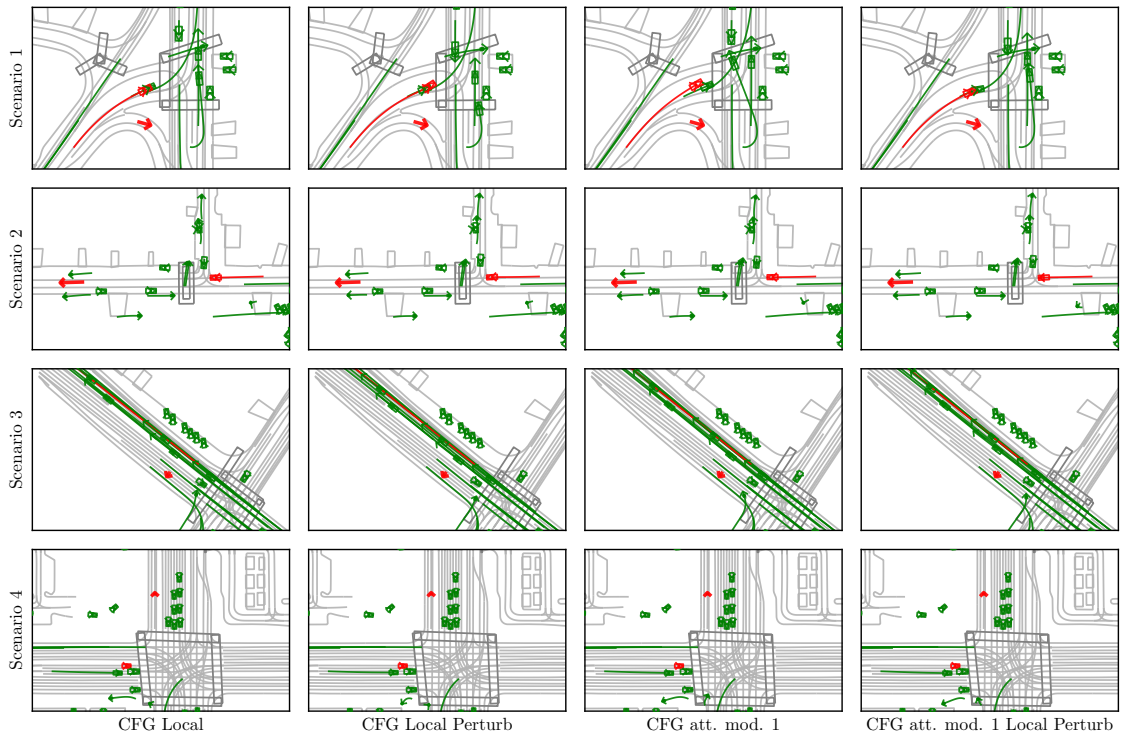


Figure A.4.1: Scenarios with trajectories generated using the final model and its predecessors with replan 80 and no guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.24	1.64	1.67	0.0938	0.111	0.108	0	0.0384	0.0709	0	0.0131	0.0418
CFG Local Perturb	1.11	1.63	2.20	0.0931	0.108	0.113	0.0318	0.0375	0.0634	0	0.0101	0.0341
CFG att. mod. 1	1.21	1.65	1.80	0.0968	0.104	0.0972	0	0.0365	0.0598	0	$8.38e{-3}$	0.0255
CFG att. mod. 1 Local Perturb	1.19	1.62	1.76	0.0960	0.108	0.103	0	0.0387	0.0894	0	0.0125	0.0399

Table A.4.2: Scenario quality metrics of background agents for the final model and its predecessors with replan 80 and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	4.31	16	22.3	5.31	16.9	22	3.76	4.38	3.44	1	1.20	0.968
CFG Local Perturb	6.05	18.6	25.1	6.99	19.4	25	3.98	4.71	3.50	1.15	1.28	0.989
CFG att. mod. 1	3.90	16.9	23.3	5.91	17.9	23.1	3.75	4.37	3.53	0.990	1.18	0.963
CFG att. mod. 1 Local Perturb	5.25	17.1	22.9	6.13	18	22.8	4.06	4.60	3.48	1.01	1.16	0.947

Table A.4.3: Target metrics for the final model and its predecessors with replan 80 and VBD guidance.

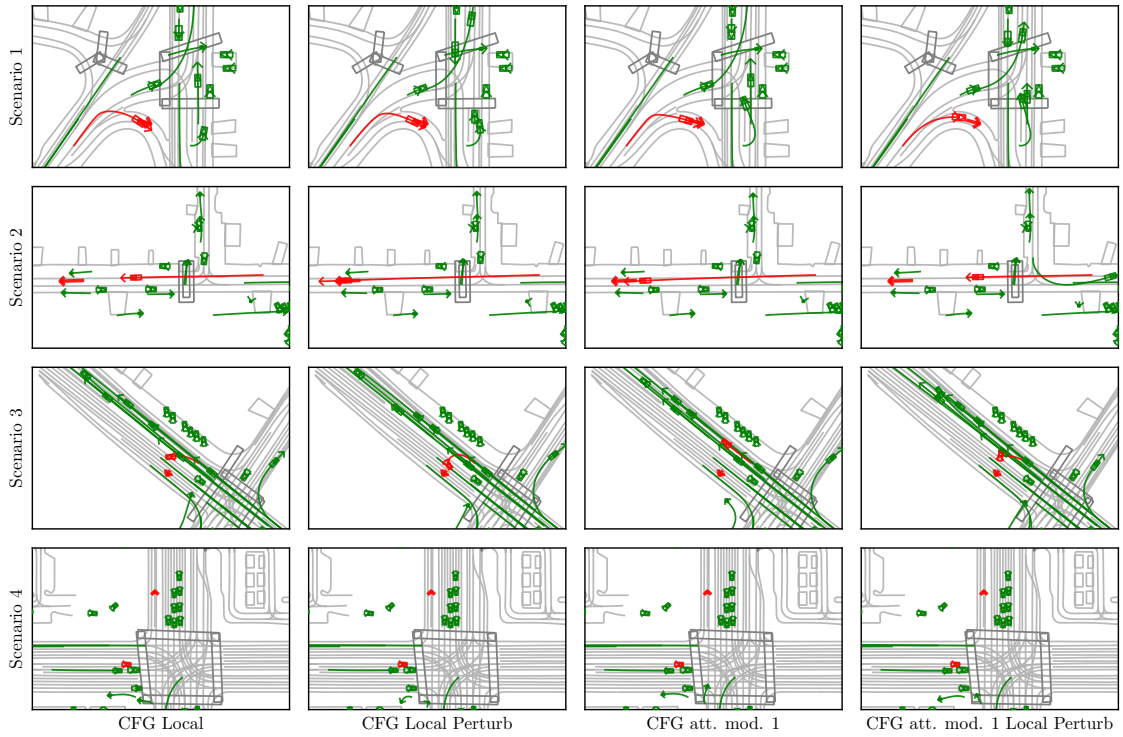


Figure A.4.2: Scenarios with trajectories generated using the final model and its predecessors with replan 80 and VBD guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.05	1.29	1.57	0.0968	0.103	0.0916	0	0.0340	0.0644	0	0.0114	0.0411
CFG Local Perturb	1.03	1.19	1.50	0.0968	0.104	0.0914	0	0.0360	0.0831	0	$8.98e{-3}$	0.0386
CFG att. mod. 1	0.980	1.18	1.49	0.0968	0.108	0.0974	0	0.0307	0.0724	0	$7.81e{-3}$	0.0270
CFG att. mod. 1 Local Perturb	0.926	1.19	1.64	0.0968	0.109	0.0964	0	0.0391	0.0919	0	0.0106	0.0288

Table A.4.4: Scenario quality metrics of background agents for the final model and its predecessors with replan 80 and CFG guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.17	5.33	8.60	8.24	11.3	11.1	0.975	1.60	1.67	0.442	0.754	0.798
CFG Local Perturb	0.858	4.30	7.74	6.75	9.73	9.99	1.24	1.91	1.91	0.596	0.844	0.829
CFG att. mod. 1	2.63	6.73	9.86	8.55	11.8	12.3	1.21	1.91	2	0.578	0.872	0.864
CFG att. mod. 1 Local Perturb	1.25	4.32	6.44	7.19	9.58	8.93	1	1.81	1.87	0.389	0.771	0.833

Table A.4.5: Target metrics for the final model and its predecessors with replan 80 and CFG guidance.

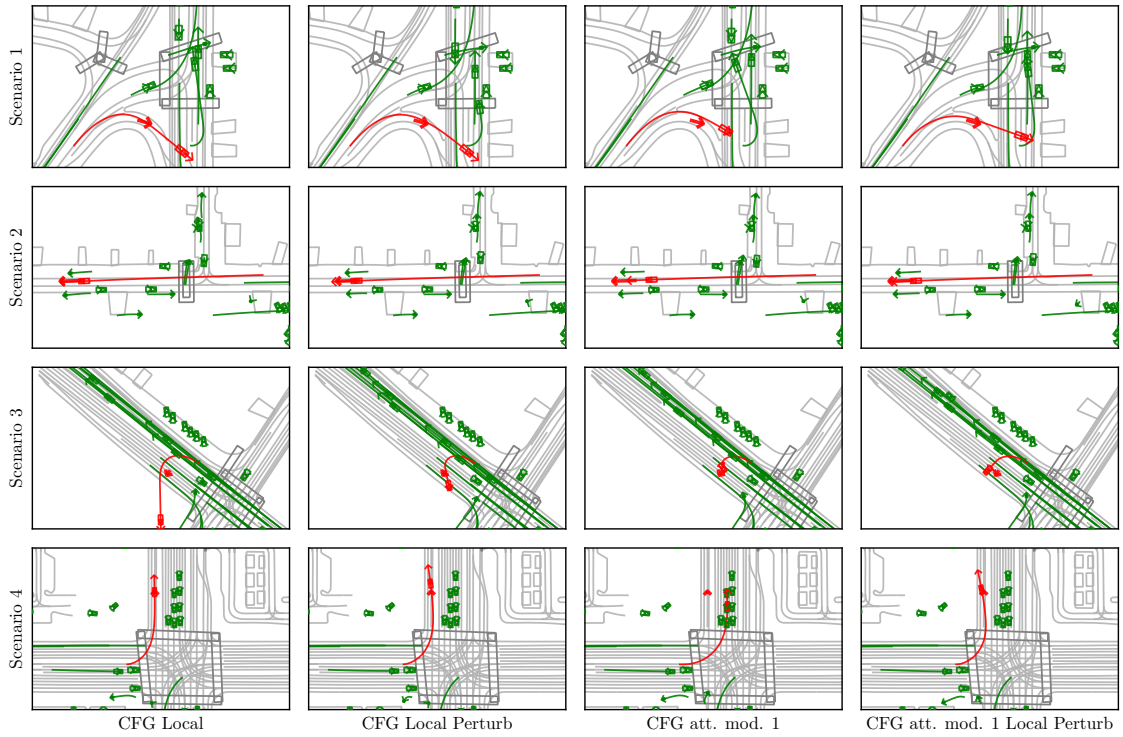


Figure A.4.3: Scenarios with trajectories generated using the final model and its predecessors with replan 80 and CFG guidance.

Model	ADE			Collisions			Offroad			Wrong Way		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	1.02	1.29	1.82	0.0968	0.109	0.0986	0	0.0342	0.0837	0	0.0107	0.0264
CFG Local Perturb	0.976	1.20	1.45	0.0968	0.105	0.0908	0	0.0361	0.0701	0	0.0101	0.0396
CFG att. mod. 1	0.995	1.14	1.60	0.0938	0.103	0.0931	0	0.0297	0.0556	0	$8.71e{-3}$	0.0242
CFG att. mod. 1 Local Perturb	0.950	1.22	1.90	0.0968	0.115	0.108	0	0.0356	0.0757	0	0.0109	0.0286

Table A.4.6: Scenario quality metrics of background agents for the final model and its predecessors with replan 80 and CFG and VBD guidance.

Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
	Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
CFG Local	0.539	1.46	3.03	0.821	1.89	3.21	0.565	1.56	1.98	0.266	0.602	0.778
CFG Local Perturb	0.593	1.97	4.97	0.954	2.38	4.96	0.724	1.58	1.98	0.326	0.645	0.775
CFG att. mod. 1	0.657	1.72	3.34	1.05	2.24	4.05	0.611	1.48	1.89	0.268	0.616	0.768
CFG att. mod. 1 Local Perturb	0.582	1	1.79	0.933	1.42	2	0.626	1.64	2.04	0.295	0.654	0.795

Table A.4.7: Target metrics for the final model and its predecessors with replan 80 and CFG and VBD guidance.

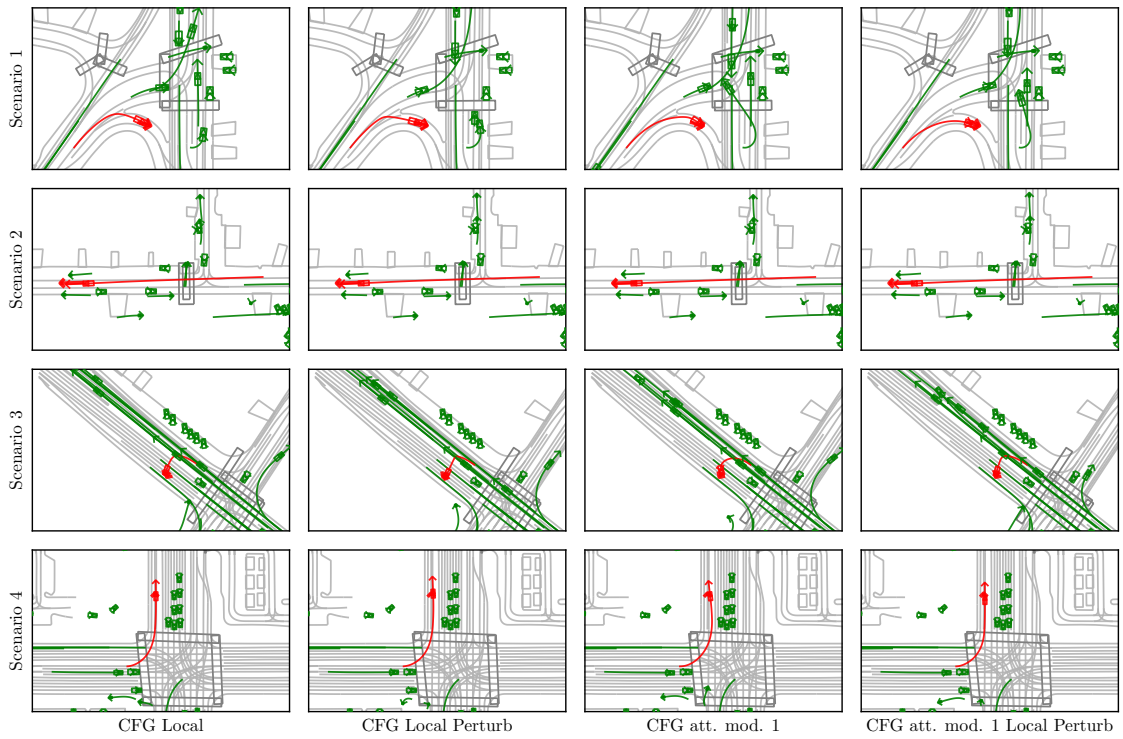


Figure A.4.4: Scenarios with trajectories generated using the final model and its predecessors with replan 80 and CFG and VBD guidance.

A.5 Guidance Methods

VBD Guid.		CFG Guid.	Model	ADE			Collisions			Offroad			Wrong Way		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG att. mod. 1	1.10	1.35	1.32	0.0645	0.0930	0.0913	0	0.0313	0.0643	0	$8.47e{-3}$	0.0244
	✓		CFG Local	1.24	1.64	1.67	0.0938	0.111	0.108	0	0.0384	0.0709	0	0.0131	0.0418
		✓	CFG Local Perturb	1.03	1.19	1.50	0.0968	0.104	0.0914	0	0.0360	0.0831	0	$8.98e{-3}$	0.0386
✓		✓	CFG att. mod. 1 Local Perturb	0.961	1.19	1.78	0.0968	0.111	0.0988	0	0.0361	0.0756	0	$8.94e{-3}$	0.0236
	✓	✓	CFG att. mod. 1 Local Perturb	0.950	1.22	1.90	0.0968	0.115	0.108	0	0.0356	0.0757	0	0.0109	0.0286

Table A.5.1: Scenario quality metrics of background agents for guidance methods with replan 80 and the models with the respective lowest minimum distance.

VBD Guid.		CFG Guid.	Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG att. mod. 1	10.2	19.2	23.9	12	21.1	23.6	4.75	5.29	3.33	1.10	1.32	0.981
	✓		CFG Local	4.31	16	22.3	5.31	16.9	22	3.76	4.38	3.44	1	1.20	0.968
		✓	CFG Local Perturb	0.858	4.30	7.74	6.75	9.73	9.99	1.24	1.91	1.91	0.596	0.844	0.829
✓		✓	CFG att. mod. 1 Local Perturb	0.756	2.16	3.69	1.82	3.33	4.32	0.947	1.81	2.01	0.346	0.715	0.793
	✓	✓	CFG att. mod. 1 Local Perturb	0.582	1	1.79	0.933	1.42	2	0.626	1.64	2.04	0.295	0.654	0.795

Table A.5.2: Target metrics for guidance methods with replan 80 and the models with the respective lowest minimum distance.

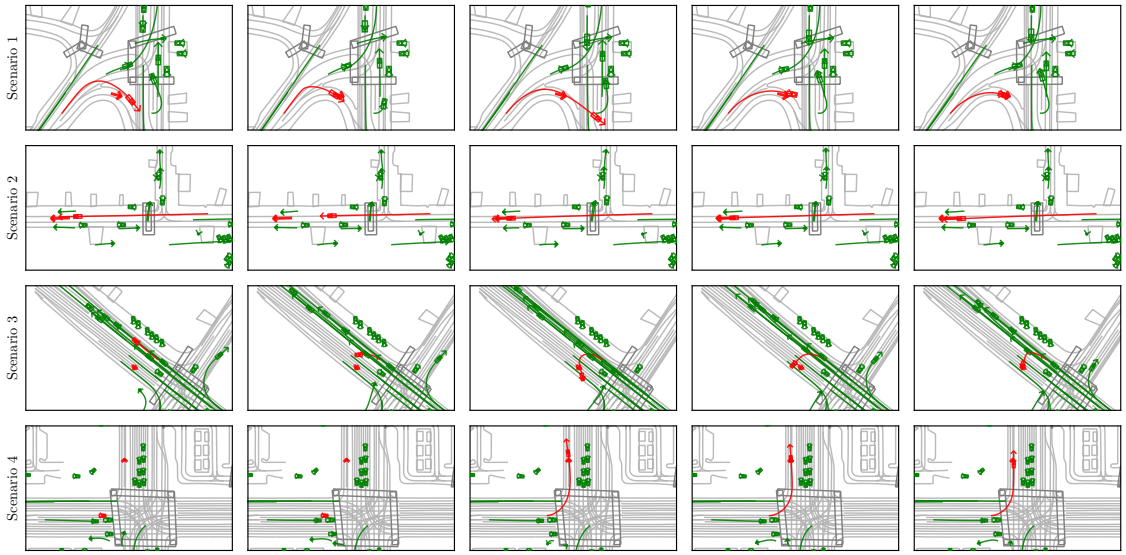


Figure A.5.1: Scenarios with trajectories generated with replan 80 using the guidance methods and the models with the respective lowest minimum distance.

A.6 Trajectory Optimization

VBD Guid.		CFG Guid.	Model	ADE			Collisions			Offroad			Wrong Way		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG att. mod. 1 Local Perturb	1.04	1.32	1.55	0.0931	0.105	0.104	0	0.0330	0.0637	0	$8.43e{-3}$	0.0246
	✓		CFG Local	1.27	1.67	1.71	0.0968	0.122	0.109	0	0.0387	0.0632	0	0.0126	0.0404
		✓	CFG att. mod. 2	0.930	1.18	1.57	0.0968	0.109	0.103	0	0.0324	0.0735	0	0.0130	0.0412
✓		✓	CFG Enc. only	0.970	1.27	1.68	0.100	0.119	0.102	0	0.0321	0.0809	0	$7.28e{-3}$	0.0224
	✓	✓	CFG att. mod. 2	0.956	1.21	1.75	0.0968	0.116	0.106	0	0.0318	0.0720	0	$8.98e{-3}$	0.0254

Table A.6.1: Scenario quality metrics of background agents for guidance methods with replan 80, trajectory optimization and the models with the respective lowest minimum distance.

VBD Guid.		CFG Guid.	Model	Minimum Distance			Final Distance			Speed Error			Yaw Error		
Iter. 1	Iter. 10			Median	Mean	std	Median	Mean	std	Median	Mean	std	Median	Mean	std
✓			CFG att. mod. 1 Local Perturb	0.808	9.91	21.1	0.968	10.1	21.1	3.24	3.82	3.43	0.820	1.04	0.936
	✓		CFG Local	0.612	8.91	19.9	0.647	9.03	19.9	1.80	3.28	3.55	0.477	0.914	0.948
		✓	CFG att. mod. 2	0.290	1.12	1.97	0.333	1.19	2.07	0.448	1.34	1.87	0.216	0.566	0.773
✓		✓	CFG Enc. only	0.290	1.12	1.81	0.328	1.20	1.93	0.349	1.28	1.90	0.200	0.542	0.763
	✓	✓	CFG att. mod. 2	0.301	1.04	1.75	0.303	1.13	1.90	0.292	1.16	1.79	0.176	0.512	0.734

Table A.6.2: Target metrics for guidance methods with replan 80, trajectory optimization and the models with the respective lowest minimum distance.

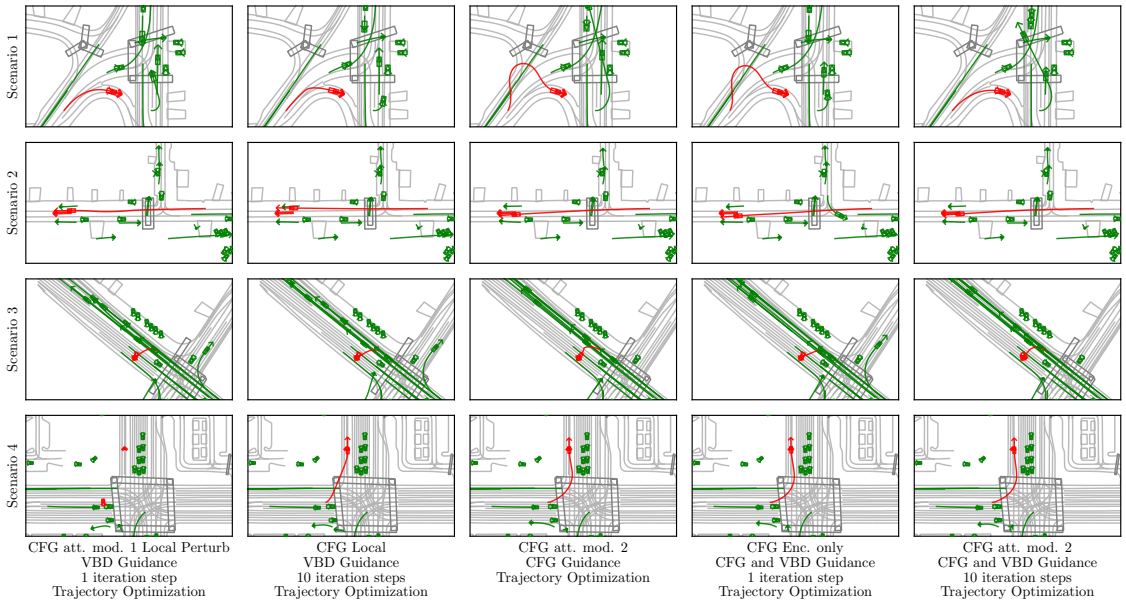


Figure A.6.1: Scenarios with trajectories generated with replan 80 using the guidance methods with trajectory optimization and the models with the respective lowest minimum distance.