Sensor simulation for autonomous vehicles: Diffusion based image and depth generation for driving scenes

by

Linn Bieske

B.S. Materials Science, RWTH Aachen University, 2015; M.S., Materials Science, ETH Zurich, 2018; M.S., Business Analytics, Imperial College London, 2018; MPA, Harvard Kennedy School, 2023

Submitted to the MIT and Sloan School of Management and
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Electrical Engineering and Computer Science

in conjunction with the Leaders for Global Operations Program

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Authored by: Linn Bieske

MIT Sloan School of Management and

Department of Electrical Engineering and Computer Science

May 9, 2025

Certified by: Rama Ramakrishnan

Professor of the Practice, AI/ML

Thesis Supervisor

Certified by: Yoon Kim

Professor of Electrical Engineering and Computer Science

Thesis Supervisor

Accepted by: Leslie A. Kolodziejski

Professor of Electrical Engineering and Computer Science

Chair, Department Committee on Graduate Students

Accepted by: Maura Herson

Assistant Dean, MBA Program MIT Sloan School of Management

Company supervisor:	Atul Kumar, mentorship by Max Jiang, Vincent Casser, Bo Sun
Areas of research:	Operations, Generative Artificial Intelligence, Autonomous Vehicles, Sensor Simulation, Scenario Generation

Sensor simulation for autonomous vehicles: Diffusion based image and depth generation for driving scenes

by

Linn Bieske

B.S. Materials Science, RWTH Aachen University, 2015; M.S., Materials Science, ETH Zurich, 2018; M.S., Business Analytics, Imperial College London, 2018; MPA, Harvard Kennedy School, 2023

Submitted to the MIT and Sloan School of Management and
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Electrical Engineering and Computer Science

in conjunction with the Leaders for Global Operations Program

Abstract

Background: Autonomous vehicle (AV) testing requires extensive real-world data collection, which is costly and time-consuming. Existing simulation techniques struggle to generate high-fidelity sensor data, particularly for multimodal signals like RGB camera images, LiDAR depth maps or LiDAR point clouds. Recent advances in generative AI, specifically diffusion models, offer a solution for improving synthetic driving scene simulations.

Objective: This thesis enhances diffusion-based generative models to: 1) Encode LiDAR depth data into a stable diffusion model's latent space, 2) Generate simultaneously, consistently and with high fidelity eight RGB camera images, 2D LiDAR depth maps and 3D LiDAR point

clouds for a full 360-degrees range, and 3) Evaluate the realism and consistency of the generated sensor data.

Methods: A multimodal, multi-view latent stable diffusion model was trained to generate complete 360’ synthetic driving scenes and simulate camera and LiDAR sensor signals for autonomous vehicles. The generated scenes were evaluated for sensor alignment, realism, and depth accuracy.

Results: The diffusion model produced realistic, spatially consistent camera and LiDAR sensor data, reducing reliance on real-world validation miles and lowering AV testing costs. To further improve the quality of the multimodal driving scene generation it is recommended to retrain the VAE on LiDAR data.

Conclusion: This work advances AV simulation by extending stable diffusion models to multimodal sensor data. Future improvements should focus on real-time generation and expanding to additional sensor types or hardware setups for enhanced simulation fidelity.

Thesis Supervisor: Rama Ramakrishnan

Title: Professor of the Practice, AI/ML

Thesis Supervisor: Yoon Kim

Title: Professor of Electrical Engineering and Computer Science

Table of contents

1 Introduction

The development of autonomous vehicles (AVs) relies on extensive real-world data collection to validate their performance and ensure safety. In addition to on-road testing, simulation-based validation presents a promising alternative, allowing AVs to be tested in diverse and complex driving conditions. While existing simulation technologies provide useful insights, a realism gap between simulated driving scenes and real driving scenes often remains. In contrast to the commonly used reconstructive scene generation techniques, recent advancements in generative artificial intelligence (GenAI) have demonstrated the potential of stable diffusion models in producing high-fidelity synthetic images and 3D perspectives. [1] However, these models have primarily been designed for visual data, indoor scenes [2] and lack the capability to generate multimodal driving scenes that accurately align LiDAR depth information with RGB images for diverse outdoor driving scenes. Therefore, we propose a novel generative framework to simulate synthetic sensor signals of autonomous vehicles for new driving scenes. We present – to our knowledge the first - multimodal, multi-view latent stable diffusion model which can generate eight camera views and the corresponding LiDAR depth maps for a full 360-degree range of an autonomous vehicle. Camera intrinsics and extrinsics are used to project the generated depth information to the 3D space such that a synthetic LiDAR point cloud is generated. Overall, our approach lays the foundation for generating synthetic camera and LiDAR sensor signals of an autonomous vehicle which could support the simulation and testing of autonomous vehicles complementing real-world data collection efforts.

The research contributions span three key areas:

1) We developed densification strategies for sparse LiDAR data to effectively encode / decode them to and from a latent space with convolutional techniques such as Variational Autoencoders (VAEs)

2) We achieved multimodal latent stable diffusion models which generate 360-degree camera views LiDAR depth maps simultaneously, with consistent perspectives and high geometric integrity of objects. The first of its kind, in the field of autonomous driving.

3) We developed a camera model which projects the generated LiDAR depth maps to 3D point clouds allowing us to simulate both camera and LiDAR sensor signals in their native formats.

Overall, this work contributes to the research field by demonstrating high-quality synthetic scene generation supporting autonomous vehicle simulation and testing. The foundations laid with this work could have the potential to reduce mileage collection efforts, operational cost and accelerate the timeline for scaling to new geographies for autonomous driving companies in the future.

To achieve the contributions summarized above, and explore the use of latent stable diffusion models multimodal sensor signal simulation for autonomous vehicles, the research was structured into four sub-questions:

a) How can LiDAR depth information be effectively encoded into the latent space of a stable diffusion model?

b) What techniques can be used to ensure spatial consistency between generated RGB images and LiDAR depth maps over a 360-degree space?

c) How can the quality and realism of generated multimodal driving scenes be evaluated against real-world benchmarks?

d) What impact does synthetic sensor data have on reducing real-world validation miles for AV testing?

These questions guided the development of modeling pipelines and planning of experimental results. Within the next sections, we summarize related work, detail the applied methodologies, report the experimental results, discuss results, and draw conclusions about our findings.

2 Related work

Recent advancements in 3D generation have led to frameworks capable of producing images and depth maps for 3D objects and camera scenes. [1] Additionally, the application of latent stable diffusion models to LiDAR data has emerged as a promising research direction, particularly in the context of autonomous driving. [3-8] This research builds upon research lines of multi-view generation, depth maps estimation and LiDAR point cloud generation. Moreover, video generation is an adjacent research field which fields promising directions for future research in the domain of sensor simulation for autonomous vehicles.

Multi-view generation. Achieving geometric consistency across multiple viewpoints in novel view synthesis remains a fundamental challenge. [1] Traditional methods such as NeRF often rely on static input views and computationally expensive volumetric rendering techniques. [9] Furthermore, neural rendering approaches, including 3D Gaussian Splatting (3DGS), face difficulties in handling occlusions and sparse input views. [10] While promising techniques have emerged that improve single-image novel view synthesis such as NerfDiff which relies on stable diffusion and introduces a NeRF-guided distillation process that addresses depth ambiguities and occlusion artifacts,[11] powerful multi-view generation frameworks have been presented such as CAT3D [1]. CAT3D introduces raymaps which encode relative camera positions (3D-aware generative model) to ensure consistent 360-degree mutli-view image and depth generation. Other diffusion models rely on training on posed camera images and depth maps to syndicate 3D environments with latent diffusion models. [12] Overall, this 3D world generation is in particularly relevant for the domain of AI agents such as autonomous vehicles. [10,12]

2D LiDAR depth map generation: Generating high-quality LiDAR depth maps remains difficult due to the inherent sparsity of LiDAR signals. Moreover, monocular depth estimation

methods often struggle with scale ambiguities and lack of geometric priors. [2,13] Vision transformers (ViTs) improve depth map prediction by maintaining global spatial coherence and outperforming traditional convolutional architectures. [13] Marigold repurposes diffusion-based image generators for monocular depth estimation, leveraging generative priors to improve zero-shot generalization and scene understanding. [2] Interestingly, literature shows that Variational Autoencoders (VAEs) trained on RGB camera images can be repurposed for depth maps and can be effectively included in generative modeling pipelines. [2] Since depth information is a critical sensor data domain for autonomous vehicles, this research will build upon this work. Furthermore, in particularly interesting for autonomous driving simulation is LidarDM which introduces a novel 4D LiDAR generative model that synthesizes layout-aware, physically plausible LiDAR depth sequences while ensuring temporal coherence. [14]

3D LiDAR point cloud generation - encoding sparse LiDAR signals into latent spaces: Simulating realistic 3D LiDAR point clouds is crucial for autonomy testing, yet traditional simulation methods often fail to capture key sensor characteristics, such as raydrop noise, occlusions, and material reflectance. [3-5] Additionally, encoding sparse LiDAR signals into latent spaces poses a significant challenge for generative modeling. [8,15] Recent advancements in LiDAR-based generative modeling have significantly improved the realism, scalability, and adaptability of synthetic sensor data for autonomous driving applications. [3-8] LidarGRIT employs an auto-regressive transformer to iteratively sample point clouds in latent space, enhancing realism compared to traditional diffusion-based approaches, thereby facilitating the creation of highly realistic 3D environments for simulation and sensor validation. [3] Addressing the domain gap in synthetic LiDAR data, *Towards Zero Domain Gap* reconstructs real-world scenarios as digital twins, providing a benchmark for validating the fidelity of LiDAR simulations and improving the robustness of perception models. [4] To bridge the gap between language and LiDAR perception, LidarCLIP introduces a zero-shot classification and retrieval

framework by mapping LiDAR embeddings into a CLIP-like space, enhancing the adaptability and generalization of LiDAR-based learning systems. [6] Additionally, LiDAR-NeRF integrates LiDAR sensor data into NeRF to encode 3D structural details, significantly improving geometric accuracy in novel view synthesis, which is essential for multi-view scene reconstruction in autonomous navigation. [13] Furthermore, SceneControl Diffusion refines traffic scene generation by modeling actor placements and interactions using a diffusion-based framework, enabling the creation of dynamic, realistic traffic environments crucial for AV behavior prediction and planning. [5]

Video generation: Achieving temporally coherent, high-fidelity video synthesis is challenging, particularly in urban scene simulation and forecasting for autonomous driving. Standard diffusion-based models often generate frame sequences with inconsistencies in object positioning and motion. [16-18] Video Diffusion Models extend image diffusion techniques to video synthesis, incorporating spatial-temporal conditioning for improved coherence. [16] For example, Streetscapes utilizes autoregressive video diffusion to generate realistic street views conditioned on maps, enabling long-range, large-scale urban simulation. [17] Moreover, Drive-WM applies multi-view forecasting to predict future driving scenes, facilitating safe trajectory planning through simulated video rollouts. [19] Moreover, real-world data is often incomplete such that Diffusion Models can play a critical role for reconstructing incomplete LiDAR sequences, preserving temporal consistency while filling in missing data. [18] Currently, our work does not generate temporal elements of synthetic sensor data. Yet, it is a promising direction for future work.

3 Method

Autonomous driving companies face significant costs associated with the test and validation miles required to safely expand autonomous ride-hailing services into new territories. Current simulation technologies still exhibit a realism gap, limiting their effectiveness in reducing these costs. This thesis addresses this gap by developing a generative model capable of producing high-quality camera and LiDAR sensor signals of an autonomous vehicle, enabling more realistic and effective driving scenario simulations. Specifically, multi-modal, multi-view latent diffusion models are developed, trained, and evaluated to generate consistent 3D camera scenes and 3D LiDAR point clouds.

First, a comprehensive dataset consisting of multi-view camera images and 3D LiDAR point clouds was extracted. Second, data transformation techniques were applied to preprocess the camera and LiDAR data. Next, the data was encoded into a latent space using a pre-trained variational autoencoder (VAE). Subsequently, stable diffusion models were implemented in Python and trained on tensor processing units. Finally, the generated camera scenes and 3D LiDAR point clouds were evaluated for their quality. These five steps will be described in more detail in the following sub-chapters. Lastly, an evaluation of the chosen research approach will be presented.

3.1 Data collection

A comprehensive dataset was collected, containing 180 TB camera images and 3D LiDAR point clouds recorded by an autonomous vehicle. The dataset spanned various geographies, times of day, and weather conditions. Eight camera views were recorded by the autonomous vehicle, while the spinning LiDAR captured 360-degree 3D point clouds, measuring the corresponding distances to surrounding objects. Approximately 1 million examples of different driving scenes were included in the dataset.

3.2 LiDAR data preprocessing

To translate the data into a suitable format for model training, key pre-processing steps were applied. This included accounting for the fact that the cameras are mounted in different positions than the LiDAR sensor, leading to varying sensor perspectives. Secondly, to apply a convolutional encoding technique, such as VAEs, to sparse data, data densification was required. Since the VAE was pre-trained on RGB camera data and repurposed for encoding LiDAR 2D depth maps, selecting an appropriate data transformation and normalization scheme was critical. This ensured the LiDAR 2D depth map values were scaled into the range of RGB images, allowing the pre-trained VAE to achieve sufficient encoding and decoding performance, making it a viable method for encoding and decoding. Finally, post-processing steps were carried out to convert the output of the generative latent stable diffusion model into a visualized driving scene. More details about the applied methods are provided in the following sections.

3.2.1 Projection

The sensing hardware of the autonomous vehicle included eight cameras capturing a 360-degree view of the driving scenes, as well as a spinning LiDAR mounted on top to capture depth information. The cameras collected 3-channel RGB images, while the LiDAR captured a 3D point cloud, with each point having three Cartesian coordinates (x, y, z).

Given that the LiDAR sensor is mounted in a different physical position than the eight cameras of the autonomous vehicle, it was necessary to geometrically align the camera views and the LiDAR 3D point cloud (see Figure 1). To achieve this, a geometric projection was implemented, which accounted for camera intrinsics, camera extrinsics, the dynamics of the rolling shutter of the spinning LiDAR, and the underlying driving dynamics.

Each LiDAR point with x, y, z coordinates was converted into a distance range and projected onto the 2D x, y matrix of the corresponding camera. Figure 2 conceptually illustrates this projection. Figure 6A and Figure 6B show a selected camera view and the corresponding projected LiDAR depth maps, respectively. It can be observed that both views are aligned.

Moreover, since the projection is based on a known set of parameters, it is reversible. Thus, it is possible to back-project from the 2D depth map (x, y, depth) representation into a 3D LiDAR point cloud (x, y, z coordinates).

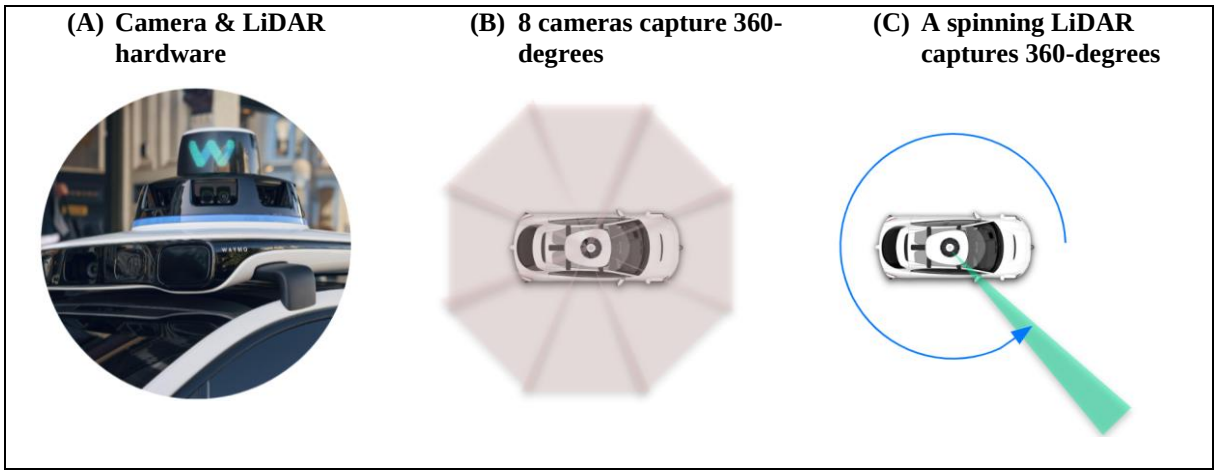


Figure 1: The camera and LiDAR sensors are positioned closely to each other. Due to the physical distance between the sensors, they have different viewing positions. (A). 8 cameras capture 8 different fields of views spanning a 360-degree range (B). A spinning LiDAR measures the distance of traffic objects based on time of flight. Through this a 3D 360-degree point cloud of the surrounding is created (C). The stable diffusion model architecture was set up such that it corresponds to this hardware configuration. Photo credits: Waymo LLC

One challenge introduced by the discrete scanning of the spinning LiDAR sensor and the projection of its signal into the camera views is data sparsity. RGB camera images are dense; in contrast, LiDAR point clouds are sparse due to the scanning frequency of the spinning LiDAR and missing returns. On the one hand, missing returns are desirable - for example, the sky does not reflect a laser signal. On the other hand, missing returns and data sparsity can be caused by scattering on sharp edges or objects with low reflectivity, such as trees. Overall, when converting the 3D LiDAR point cloud into eight 2D depth maps consistent with the RGB camera views, the resulting matrices are sparse. Figure 2 visualizes the 3D and 2D data

representations of LiDAR sensor values and conceptually illustrates the sparsity in the resulting 2D matrix (2D LiDAR depth map). Figure 6B shows a sparse 2D LiDAR depth map.

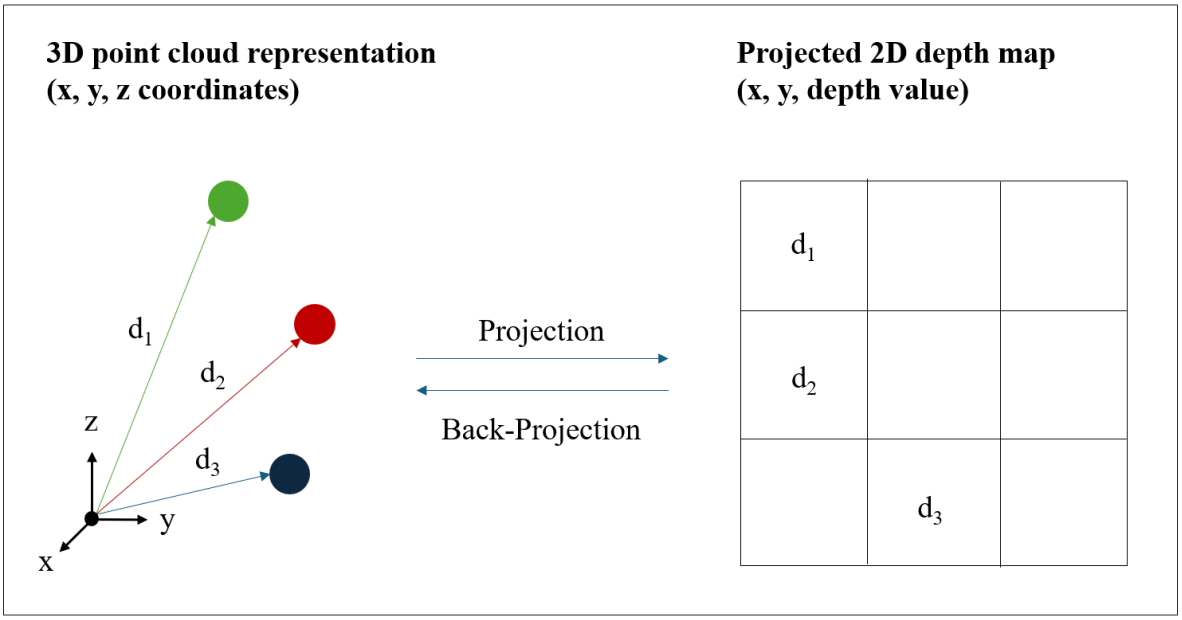


Figure 2: The LiDAR point cloud representation is converted from 3D x, y, z coordinates into depth values which are projected into the 2D camera views. Camera intrinsics, camera extrinsics, dynamics of the spinning LiDAR shutter and driving dynamics inform the geometric model to translate between the LiDAR point of view and the camera point of views. Given the underlying sparsity of the LiDAR point cloud the resulting 2D depth map is also sparse.

Unfortunately, data sparsity compromises the viability of convolutional encoding methods such as VAEs, as arithmetic calculations like averages become skewed. To address this, densification techniques were applied to the projected 2D LiDAR depth maps to achieve sufficiently high encoding and decoding consistency for the LiDAR sensor data. Different data densification approaches were investigated, and the employed methodology is described in the next section.

3.2.2 Data densification

Zero values or missing values can disrupt convolutional encoding techniques and lead to poor encoding and decoding performance. For example, zero values skew average calculations when applied to a patch of the 2D depth map, while missing values reduce the information density carried by a matrix patch, introducing instabilities and noise into the latent space. Therefore, strategies were developed to address both:

1. **Handling missing returns**, distinguishing between desirable and undesirable missing returns.

2. **Handling zero values** (invalid return values).

Two densification techniques were implemented, tested, and evaluated:

- **Max pooling:** kernel size 3x3, stride 1, padding 1.

- **K-nearest-neighbor (kNN) median pooling with zero exclusion:** kernel size 3x3 or 5x5, stride 1, padding 1.

The application of max pooling resulted in missing returns being projected to the background. For example, LiDAR points missing around sharp edges were projected to the background, causing the background to be "prioritized" by this densification technique. A small kernel size was intentionally selected to base the filling mechanism on very local information, preserving the sharpness of detected objects. To fill larger areas with missing values, max pooling was applied iteratively (1–8 times). The impact of this densification technique and the number of iterations on the VAE's encoding and decoding performance was studied. To evaluate performance, input and output 2D LiDAR depth maps or 3D LiDAR point clouds were compared, for instance, by overlaying them. The findings of this study are presented in the results section.

This first approach successfully filled missing values but may have contributed to reduced quality in encoding and decoding sharp edges, as LiDAR points were projected to the background, potentially causing edge smearing. To address this, an alternative approach was implemented: **kNN median pooling**, which excluded zeros from the median calculation. This method was chosen to balance the prioritization between projecting points to the foreground (min pooling) and the background (max pooling). Excluding zeros ensured that the median

calculation more accurately reflected the actual distances of objects with sharp edges. Two kernel sizes were tested: 3x3 and 5x5, to assess the trade-off between patch size and the required number of iterations.

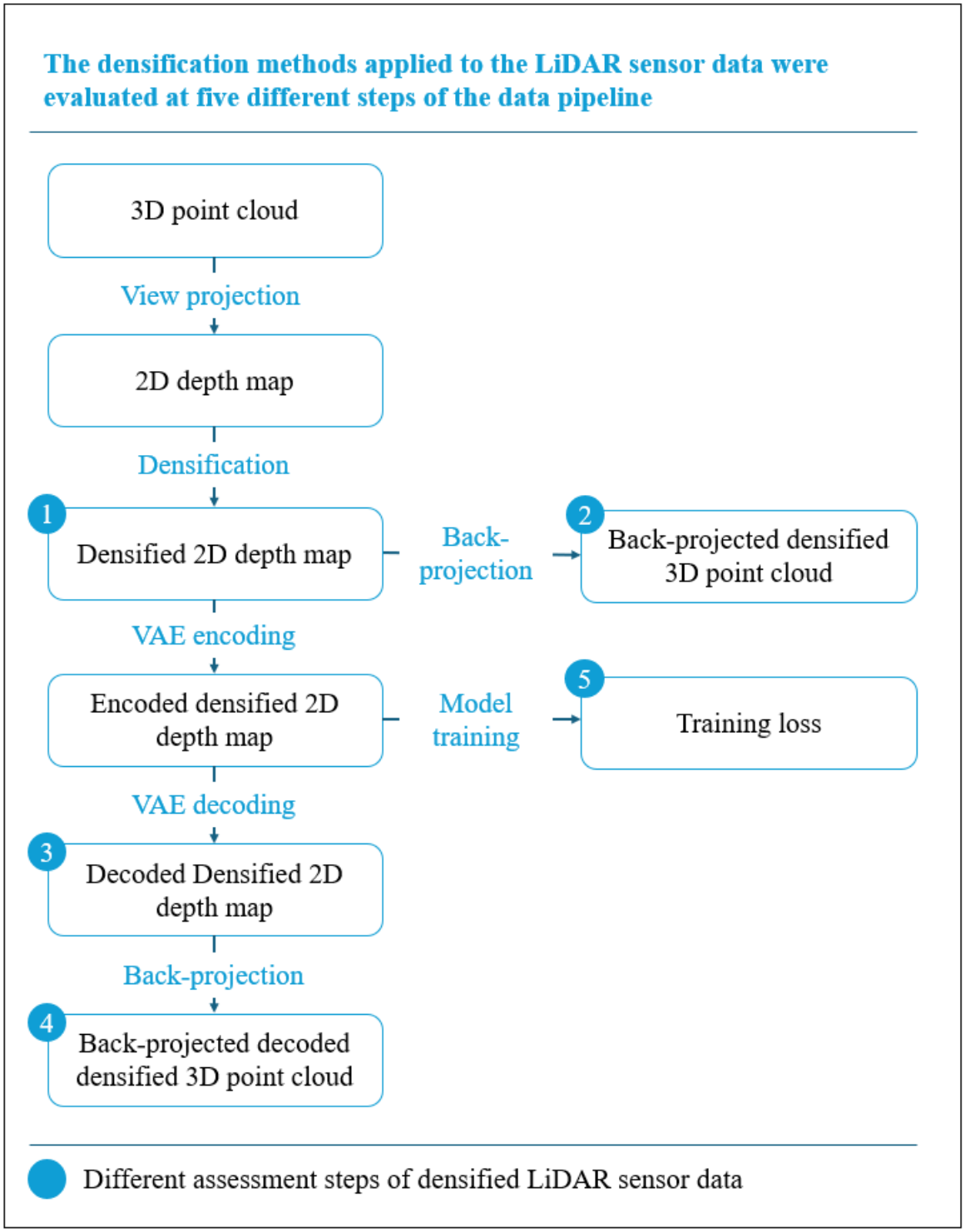


Figure 3: Along the data pipeline the performance of the LiDAR data densification methods was assessed within five steps.

Both techniques only filled missing or invalid values without altering valid ones. This was achieved by applying a validity mask.

The evaluation of LiDAR data densification was conducted across four different stages of the data pipeline. Figure 3 provides an overview of the data pipeline, which projects 3D LiDAR point clouds into 2D depth maps, applies data densification methods, encodes the 2D depth maps into a latent space, and subsequently projects the decoded densified 2D depth maps back into the 3D space as LiDAR point clouds.

This data pipeline was used to test and evaluate different data densification methods and assess both the encoding and decoding performance of the VAE, as well as the performance of a trained latent stable diffusion model in generating RGB camera views and 2D LiDAR depth maps, which are then projected into 3D LiDAR point clouds.

Five analysis steps were performed to assess the different LiDAR densification schemes. Figure 3 illustrates these steps:

1) **Visual inspection of the densified 2D LiDAR depth map**: The densified 2D depth map was visually inspected to identify remaining holes and evaluate the geometric integrity of the driving scene.

2) **Visual inspection of the 3D projection of the VAE input**: The densified LiDAR depth map was projected into 3D as a point cloud and visually inspected for its geometric integrity.

3) **Quantitative VAE performance assessment**: The densified 2D LiDAR depth map was encoded and subsequently decoded using the VAE. The L1-norm between the VAE input and output was calculated for valid pixel values to quantify the VAE's encoding and decoding error.

4) **Visual inspection of the 3D projection of the VAE output**: The densified VAE output was filtered for valid pixels, projected into 3D as a point cloud, and visually compared to the reference (raw) point cloud.

5) **Quantification of improvement in model training loss**: Different latent stable diffusion models were trained using the same model architecture but different input datasets. These datasets were prepared by applying either max pooling or kNN median pooling with zero exclusion as the densification algorithm, varying the number of iterations.

The assessment results will be presented in the results section.

3.2.3 Field of view adjustment

The LiDAR field of view is significantly narrower in the vertical dimension, as the height of the LiDAR beam is limited. It is assumed that better model training performance can be achieved when the LiDAR field of view has clean edges; therefore, the LiDAR 2D depth maps were masked to the specific field of view. Across the eight camera views, the height of the LiDAR field of view varied, and this variation was taken into account.

3.2.4 Data normalization

The VAE was pre-trained on RGB data and expects matrices distributed in the range of 0 to 1. Therefore, RGB camera data was linearly normalized into this range.

Literature has shown that VAEs trained on RGB images can be repurposed to encode and decode LiDAR depth maps, and subsequently used in the data pipeline of a generative stable diffusion model. Fundamentally, this approach only works if the underlying data distributions of RGB images and LiDAR depth maps are highly similar. As a result, range clipping and

normalization techniques played a critical role in transforming LiDAR 2D depth map data into the 0 to 1 range and ensuring its distribution closely resembled that of RGB camera images. Although LiDAR sensors have a foresight of several hundred meters, most depth values fall below 150 meters. Consequently, the LiDAR depth data was clipped between 0 meters and 150 meters.

Two different normalization schemes were applied:

- **Linear Min–Max Normalization** (Equation 1)
- **Non-linear tanh normalization**, including a scaling factor

The linear normalization preserves the geometric relationships of the point cloud in a linear manner. However, encoding the entire data range between 0 meters and 150 meters with high quality may remain a challenge. Linear normalization might compress the entire data distribution into a narrow range, causing it to differ significantly from the distribution of standard RGB camera images.

Equation 1: max-min normalization

$$x' = \frac{x - min}{max}$$

In contrast, non-linear tanh normalization was applied to address potential limitations in encoding the large dynamic range of LiDAR sensors. A large dynamic range refers to the ability to capture objects situated both in close proximity to the autonomous vehicle and at far distances. For example, this includes a neighboring vehicle as well as a vehicle located farther away. Due to the shape of the tanh function, data within the 0 m to 150 m range is distributed more evenly across the 0 to 1 range compared to the linear Min–Max normalization scheme.

Equation 2: scaled tanh normalization

$$x' = \tanh(x \cdot s)$$

LiDAR sensors are hardware components of the vehicle that support long-distance perception for autonomous vehicles. To make synthetic sensor data a viable option for the testing and validation of autonomous vehicles, both the near range and far range should be generated with sufficient geometric integrity.

It is expected that the "dynamic range" of a driving scene and the chosen normalization scheme impact both the performance of the VAE encoding and decoding processes, as well as the training loss of the trained latent stable diffusion model and the corresponding generated output. Therefore, two sets of experiments were conducted:

a) Investigating the impact of distance ranges and normalization schemes on VAEs

b) Researching the achieved model performance for different training data normalization schemes

3.3 Encoding and decoding with Variational Autoencoders

To train a latent stable diffusion model, the training data must be transformed from its high-dimensional space into a lower-dimensional latent space, for example, from 512 × 512 pixels to 64 × 64 pixels. Representing training data in a compressed latent space provides several benefits for the expected training performance of a stable diffusion model. [20] Specifically, compressing high-dimensional data, such as LiDAR and camera inputs, into a lower-dimensional latent space reduces noise, retains essential features, and enhances information density. [20] To achieve this, an encoding scheme must be selected that aligns with the architecture of the latent stable diffusion model to be trained.

Table 1: Comparison of common encoding technique based on [21]

Model	Benefits	Drawbacks
Autoencoders (AEs)	• Efficient for feature extraction and denoising • Simple architecture • Fast training	• Unstructured latent space • Poor generative capabilities • No probabilistic element
Variational Autoencoders (VAEs)	• Structured latent space • Probabilistic modeling • Can generate new data • Computationally efficient	• Lower sample quality compared to GANs • Posterior approximation can be inaccurate
Normalizing Flows (NFs)	• Exact likelihood estimation • Flexible distribution modeling • Suitable for density estimation	• High computational complexity • Requires invertibility • Harder to train

Different encoding schemes are available, such as Autoencoders (AEs), Variational Autoencoders (VAEs), and Normalizing Flows (NFs). [20-22] Table 1Table 1 summarized the benefits and drawbacks of each encoding technique. Unlike AEs, which can produce disorganized latent spaces and struggle with data generation due to the lack of a probabilistic element, VAEs maintain structure, reduce sensitivity to noise, and map data probabilistically into the latent space. [20,21] Compared to NFs, VAEs are more computationally efficient and easier to train, while still providing strong generative capabilities. This makes VAEs ideal for complex data modeling and simulation tasks. [21] Therefore, a Variational Autoencoder was chosen to encode data into a lower-dimensional latent space and subsequently decode generated model results from the latent space back into the high-dimensional input space.

To develop a latent stable diffusion model that supports both data domains—eight camera images and eight LiDAR depth maps—the VAE also needed to perform sufficiently well for both domains. Each data domain was encoded separately and subsequently concatenated to form the multi-modal training input for the latent stable diffusion model. Following the same principle, the jointly generated camera and LiDAR latents were split into a latent vector for the camera images and a latent vector for the LiDAR 2D depth maps, which were then separately decoded by the VAE.

A comprehensive assessment was conducted to analyze the performance of a VAE trained on camera images for the encoding and decoding of LiDAR 2D depth maps. The findings of this analysis will be presented in the results section.

3.4 Latent stable diffusion models: architecture and training

The goal is to train a latent stable diffusion model capable of generating synthetic sensor signals for new driving scenes. Simulating synthetic sensor signals for the cameras and LiDAR sensors of an autonomous vehicle could potentially enable the testing and validation of autonomous vehicle behavior and performance. By bringing testing from the road into a simulated 3D environment, operational costs for testing and validation are expected to decrease. Furthermore, highly realistic digital simulations that rely on sensor signals would not only be first-principle-driven but could also accelerate testing and validation timelines through improved simulation realism. A latent stable diffusion model, which relies on a transformer architecture, was chosen as the generative model for this task.

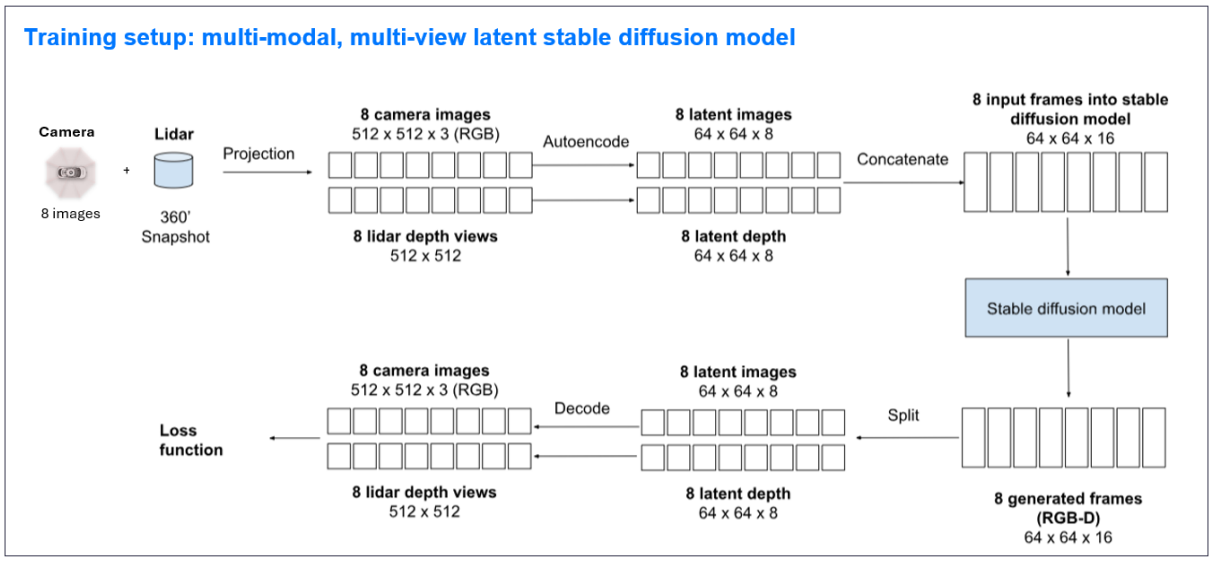


Figure 4: Data pipeline for training a multi-modal, multi-view latent stable diffusion model which can generate eight camera views and eight 2D LiDAR depth maps which get projected to a 360-degree 3D LiDAR point clouds.

A Latent Stable Diffusion Model is a generative model designed to create high-quality images by operating in a compressed, lower-dimensional latent space rather than the high-dimensional pixel space. To set up a latent stable diffusion model architecture capable of generating multiple views and multiple data domains simultaneously, the dimensionality of the latent space was set to 64 × 64 × 16: eight camera views and eight LiDAR depth maps were concatenated into 16 dimensions. Each camera view or LiDAR depth map matrix has a dimensionality of 64 × 64 pixels with three channels. For the camera views, the three channels capture RGB information. To match this channel dimensionality, LiDAR depth maps were duplicated three times, resulting in redundant information. In the future, the three channels could carry different information, though the correlation between these channels would need to be carefully considered.

In addition to the 16 latent channels (eight camera views + eight LiDAR depth maps), camera positions were added to the latent space as a "raymap" along the channel dimension. The raymap encodes the ray origin and direction at each spatial location, representing the camera pose. It also has a 64 × 64-pixel size to match the latent space dimensionality. The camera pose of the first view was chosen as the reference view, with the rays of the remaining seven camera views calculated relative to this. These 3D world coordinates condition the latent stable diffusion model on the eight different camera poses, assembling a complete 360-degree perspective for both the camera and LiDAR images.

The developed multi-modal, multi-view latent stable diffusion model was trained on 180 TB of training data, containing diverse driving scenes. The training was conducted on tensor processing units (TPUs), enabling a high level of computational parallelization and accelerated

model training. Seven different model versions were set up, trained, and evaluated. Detailed results will be presented in the results section.

To train the latent stable diffusion model the following noise schedule and loss function were implemented: At each training step of the diffusion model, a clean camera image and LiDAR depth map are selected, encoded into the latent space and concatenated to the vector $\mathbf{x_0}$. Moreover, a timestep **t** is sampled uniformly from the range [1, T]. Gaussian noise $\boldsymbol{\varepsilon \sim \mathcal{N}(0, I)}$ is then added to the latent vector to generate a noisy version $\mathbf{x_t}$ according to the forward diffusion process:

$$\mathbf{x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{(1 - \bar{\alpha}_t)} \cdot \varepsilon} \quad (1)$$

The parameter $\boldsymbol{\bar{\alpha}_t}$ denotes the cumulative product of the noise schedule parameters up to timestep **t**. The objective of the model $\boldsymbol{\varepsilon\_\theta(x_t, t)}$ is to predict the noise $\boldsymbol{\varepsilon}$ that was added to $\mathbf{x_0}$, given only the noisy image $\mathbf{x_t}$ and the timestep **t**. Note, the loss function utilizes the noise at each time step instead of the actual latent vector.

The training loss, referred to as the simplified loss, is defined as the expected mean squared error between the model's predicted noise and the actual noise used in the forward process:

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

This loss function performs an element-wise squared difference between the predicted and true noise, sums over all dimensions, and averages the result. By minimizing this loss, the model learns to accurately estimate the noise added at each timestep, thereby enabling it to reverse the diffusion process and progressively denoise samples during generation.

After training the models for multiple days, the model inference pipeline was implemented. Prompted by a random seed, the latent stable diffusion model generates a data frame that includes eight camera images and eight LiDAR depth maps (Figure 5). To decode each data

domain separately, the generated latent vector is split into a camera view vector and a LiDAR 2D depth map vector. A pre-trained VAE is used for decoding. The eight decoded camera views are assembled into a 360-degree panoramic image, simulating the camera sensor input. Additionally, the decoded 2D LiDAR depth maps are projected into 3D space as LiDAR point clouds. To achieve this, default settings for the camera extrinsics and intrinsics were chosen based on log data. This information is critical for enabling the projection from the 2D to the 3D LiDAR space.

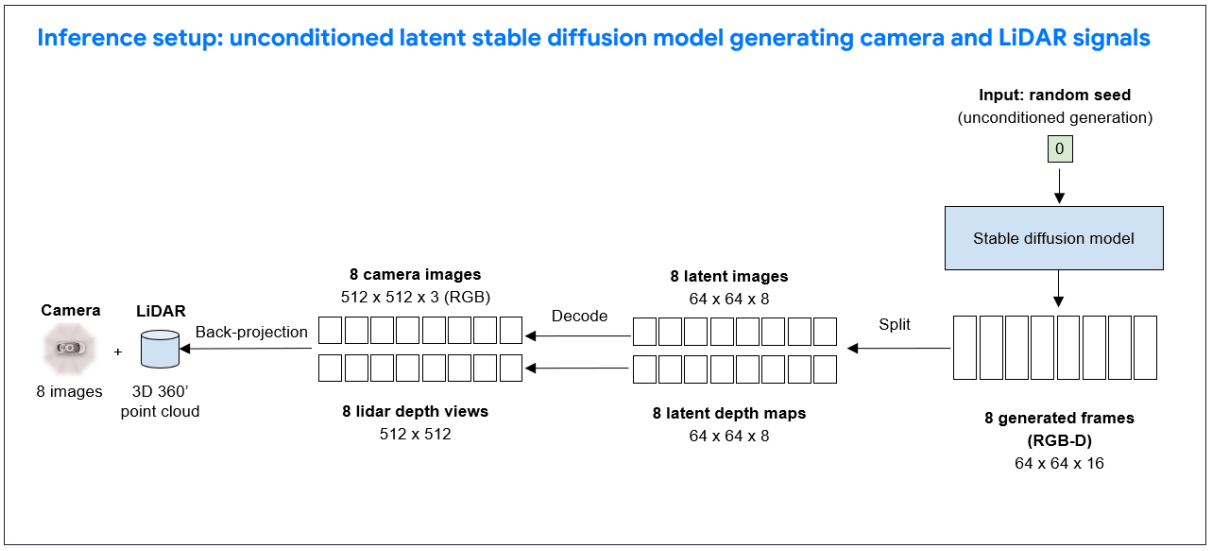


Figure 5: Inference setup for trained latent stable diffusion models. Prompted with a selected random seed, the model generates eight camera views assembling to a 360-degree panoramic image and eight LiDAR depth maps with are projected to a 3D LiDAR point cloud. Therefore, the generative latent stable diffusion model can be used for simulating the camera and LiDAR sensor signals for synthetic driving scenes.

Overall, as the generation of camera images and LiDAR 2D depth maps is prompted solely by a selected random seed, the model is unconditional; it does not rely on text prompts, input images, or additional information. In the future, the model could be expanded to include conditioning on various input prompt formats, such as scene descriptions, time of day, weather conditions, bounding boxes for traffic objects, or other contextual details. To train conditioned stable diffusion models would require to concatenate additional data vectors into the latent space and through this train the model on the joint probability distribution. When prompting the model, instead of just providing a random seed, additional model inputs are required such

as a text prompt, an image of a driving scene, weather conditions, the time of day or bounding boxes of traffic objects. Through this data generation can be controlled more precisely and steered in desired directions which are in particularly relevant for the performance testing and validation of autonomous vehicles.

3.5 Model evaluation

The trained generative latent stable diffusion models were evaluated based on both quantitative and qualitative criteria. First, the model loss was calculated for both the training and test datasets. This loss was tracked over time and compared across different model versions. Second, synthetic driving scenes were generated by prompting the models. The generated results were visually inspected and evaluated for completeness, geometric consistency, and alignment across the two data domains. For the generated LiDAR sensor signals, this evaluation was conducted in both 2D and 3D spaces.

Based on the generated results, opportunities for improvement were identified, and additional model versions were implemented, tested, and evaluated. These improvements primarily focused on data pre-processing and targeted areas such as LiDAR data densification approaches, masking data to achieve a clean LiDAR field of view and utilizing non-linear depth normalization. Additionally, the number of training steps was increased to enhance model performance.

4 Results

4.1 Qualitative assessment of LiDAR data densification approaches

A comprehensive visual inspection was performed to qualitatively assess the performance of the developed data densification approaches. The evaluation of sparse and densified LiDAR point clouds was carried out in both the 2D and the 3D space for which the results will be presented in the next sections.

4.1.1 Evaluation of densified 2D LiDAR depth maps

Figure 6 compares a dense camera view (A), a corresponding sparse LiDAR 2D depth map (B), and a densified LiDAR 2D depth map (C) obtained after applying one iteration of kNN median pooling. As shown in the example, most missing values in the 2D LiDAR depth map were successfully filled. However, assessing the quality of the data densification approach in 3D is also crucial. To this end, the densified 2D depth maps were projected back into 3D, and their geometric properties were analyzed (see Figure 7).

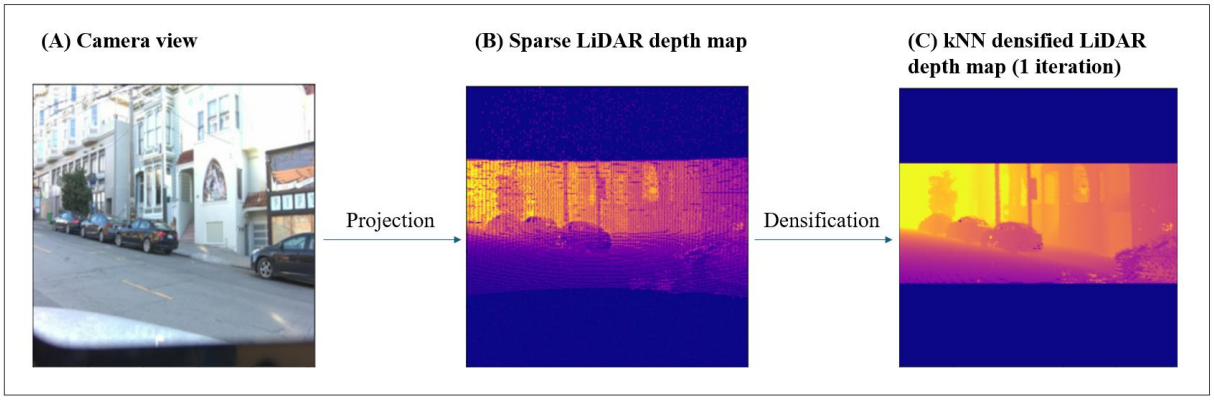


Figure 6: The projection of the LiDAR point clouds into the camera views (A) leads to a sparse LiDAR depth map (B). To effectively apply convolutional encoding techniques such as VAEs densification is required since missing data points, or zero values will break convolutions. For example, k-nearest-neighbors median pooling is applied to densify the 2D LiDAR depth map (C).

4.1.2 Evaluation of densified 3D LiDAR point clouds

The densified LiDAR depth map was projected into 3D as a point cloud and visually inspected for geometric integrity.

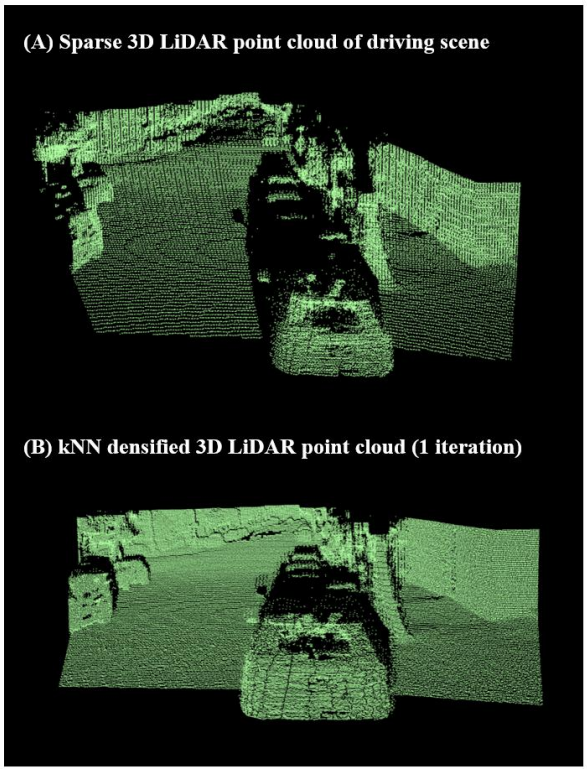


Figure 7: Projected 3D LiDAR point clouds for raw and densified 2D depth maps. The densified 2D depth maps lead to densified 3D LiDAR point clouds where objects exhibit more "block-like" appearances. Visual inspection indicates that the geometric integrity of the driving scene is well-preserved.

Figure 7 demonstrates that the geometric features of the densified 3D LiDAR point cloud are preserved, confirming that the densified 2D depth maps are a viable input for a VAE and the training of a generative latent stable diffusion model. Despite careful visual inspection, distinguishing differences in the geometric integrity of 3D point clouds generated from densified 2D LiDAR depth maps with 3×3 or 5×5 kernel sizes remains challenging. Consequently, the evaluation will be conducted in downstream stages of the data pipeline, focusing on the performance of the VAE, the training of the latent stable diffusion model, and the analysis of the achieved training loss values. The experimental results will be presented in the next section.

4.2 Evaluation of Variational Autoencoder

The densified 2D LiDAR depth maps were encoded and subsequently decoded using the Variational Autoencoder (VAE), which had been pre-trained on RGB camera images. We performed no additional training on the VAE. To adapt the VAE to a new data domain—LiDAR

depth maps—we tested and evaluated seven different data normalization approaches in experiments 01 - 07.

Since the VAE was originally trained for RGB camera images, we paid special attention to its efficacy with LiDAR depth maps. The performance assessment of the VAE on LiDAR data is structured using a top-down logic; initially, aggregated error values for a single camera (front camera) are analyzed. This is followed by the presentation of error distribution curves for all eight camera views. Finally, we conducted a detailed visual inspection in 3D for each experiment and selected camera views, enabling a qualitative performance assessment of the VAE. This inspection also included comparing the alignment of 3D point clouds between the reference and the decoded outputs.

4.2.1 Quantitative performance assessment

The following error metrics were defined and used for the qualitative performance assessment of the VAE for the LiDAR data domain:

1. **L1-norm error**: To quantify the encoding and decoding error, the L1-norm between the VAE input LiDAR depth map pixels and the VAE output LiDAR depth map pixels was computed for valid pixel values. Based on this, absolute and relative aggregated error metrics were calculated, such as absolute mean error, relative absolute mean error, absolute median error, and relative absolute median error.

2. **Error distribution**: To understand the error distribution, it was calculated what percentage share of data points have errors below a specific distance deviation, for example, within a range of plus/minus 0.5 meters. Of course, an autonomous vehicle needs to have precise distance metrics to avoid colliding with objects. Therefore, it is crucial to inspect the depth errors introduced by the VAE in more detail.

Details about the experimental setup and chosen data densification and normalization approaches are summarized in Table 2. Additionally, key error metrics are listed for the front camera. While initially the absolute mean error between the reference LiDAR depth map and the VAE encoded/decoded output depth map was 6.25 m (21.22%) for an experiment, improved performance can be unlocked by introducing data densification strategies which ensure that the convolutional elements of VAEs produce meaningful results. Densifying the sparse LiDAR depth maps reduced the absolute mean error to 1.26 m (4.17%) as experiment 02 shows. When limiting the depth range to 30 m (experiment 03) to assess the performance of the VAE in the short distances range, the absolute mean error can be reduced to 0.25 m (1.36%). This shows that the RGB pretrained VAE is transferable to the near range LiDAR depth domain. 92.8% of all LiDAR data points have an absolute error below 0.5 m, which attests to strong VAE performance in the near range. This finding points out that, contrary to claims in the literature, it remains a challenge to repurpose a VAE trained on RGB camera images for the encoding and decoding of LiDAR depth images across a large distance range (150 m). For an autonomous vehicle, LiDAR technology is crucial for its long-distance foresight. Additionally, objects can be in very close proximity to the LiDAR sensor of an autonomous vehicle, as other cars can be in the lane right next to the autonomous vehicle. Consequently, a VAE needs to be able to encode and decode a large dynamic range with sufficient accuracy to be a viable technique for building a latent stable diffusion model which simulates the camera and LiDAR sensor signals for realistic driving scenes. To achieve this capability, non-linear normalization techniques were explored in contrast to linear normalization. Tanh functions were used to rescale the depth data into a range between 0 and 1, enabling a reduction in the relative absolute median error and, in particular, increasing the percentage share of LiDAR data points having errors below 0.5 m. While the absolute median error did not further decrease, it can be seen that it is critical to go beyond aggregated error metrics for a more in-depth understanding of what drives the average value. While Table 2 only shows the error metrics for the front camera, the non-linear

normalization scheme most dominantly improved the performance of the side cameras, which were initially having inferior performance. Figure 8 provides an in-depth analysis of the error distribution curves for all eight cameras and all seven experiments, allowing a full performance comparison.

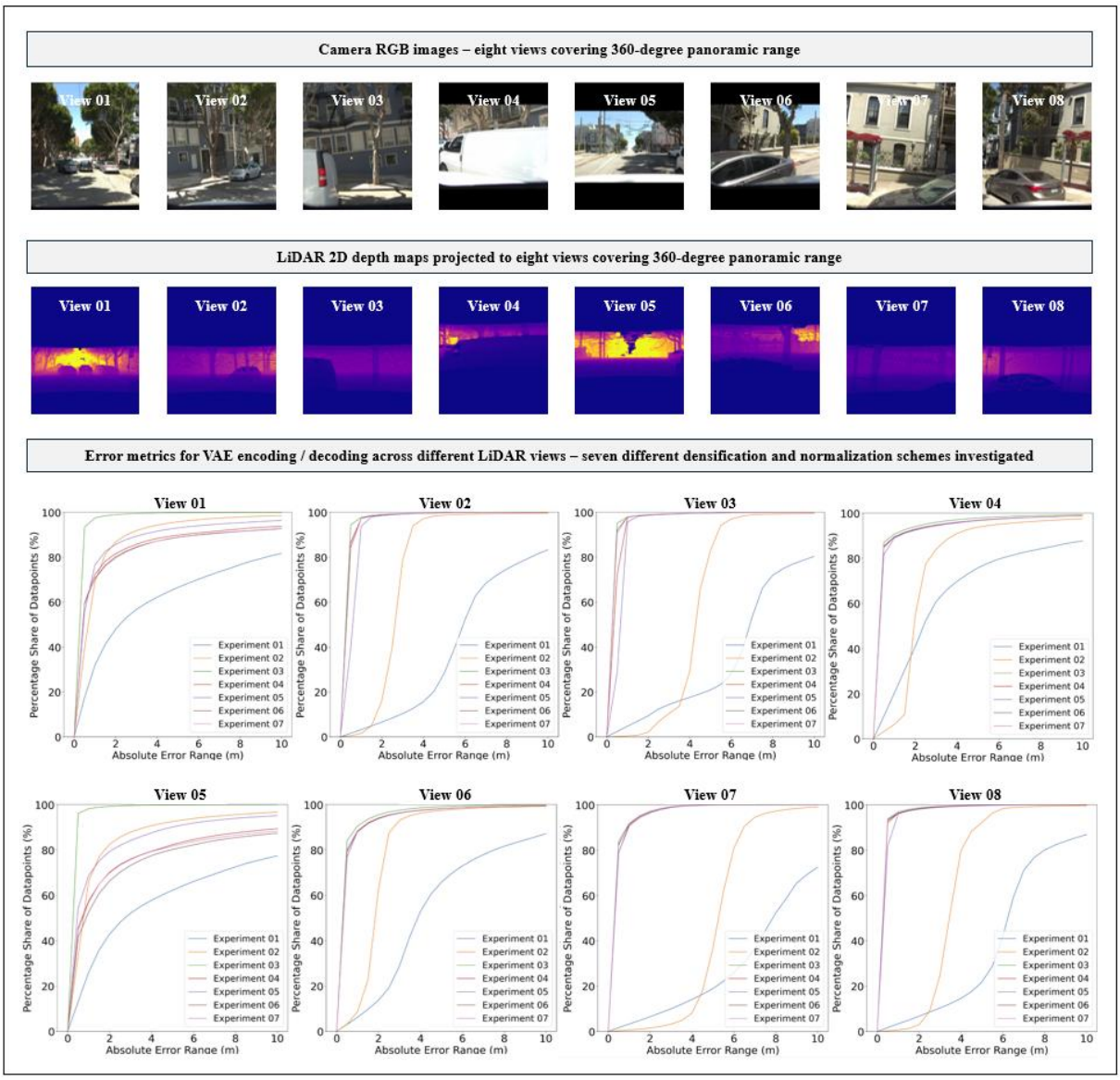


Figure 8: Assessment of the performance of a RGB pre-trained Variational Autoencoder for the encoding and decoding of 2D LiDAR depth maps.

Table 2: Seven different experiments were conducted to encode sparse LiDAR depth maps using Variational Autoencoders into a latent space and evaluate the decoded 2D LiDAR depth maps against the reference ground truth using L1 norm error metrics. The table summarizes details of the investigated data normalization and densification approaches and presents quantitative results for the front camera of an autonomous vehicle (view 01).

Experiment	**Normalization and densification approach**	**Error metrics front camera (view 01)**		
		Abs. mean error	**Rel. abs. median error**	**%share of errors within +/- 0.5 m**
01	**No densification** **Linear normalization** Clipping to 0, 150 m	6.25 m	21.22 %	17.03 %
02	**2x densification** **Linear normalization** 2x MaxPooling* Clipping to 0,150 m	1.26 m	4.17 %	39.04 %
03	**2x densification** **Linear normalization** MaxPooling* **Near range clipping** Clipping to 0, 30 m	0.25 m	1.36 %	92.8 %
04	**2x densification** **Tanh normalization** Tanh factor: 0.03 2x MaxPooling* Clipping to 0, 150 m	2.63 m	4.09 %	59.0 %
05	**2x densification** **Tanh normalization** Tanh factor: 0.02 2x MaxPooling* Clipping to 0, 150 m	1.58 m	3.56 %	53.9 %
06	**2x densification** **Tanh normalization** Tanh factor: 0.03333 2x MaxPooling Clipping to 0, 150 m	3.13 m	4.51 %	59.11 %
07	**8x densification** **Tanh normalization** Tanh factor: 0.03333 8x MaxPooling* Clipping to 0, 150 m Avg. all channels	4.27 m	1.87 %	58.96 %

3. Note: * Kernel 3x3, stride 1

When analyzing the error curves in Figure 8 the following aspects stand out:

1. **Data densification is critical:** Comparing experiments 01, 02, and 03 shows that significant improvement is achieved for all eight cameras, as the curves are shifting towards the left top corner of the graphs in Figure 8. In conclusion, applying data densification to sparse LiDAR depth maps is a critical element for achieving high VAE encoding/decoding performance.

2. **Performance gap across cameras can be closed:** When comparing the error curves for all eight camera views, it shows that it initially was challenging to achieve equally good performance across all perspective angles. For experiment 01, the front and rear cameras had smaller errors compared to the side cameras. By improving the experimental setup, it was possible to close the gap in performance between different camera views. This lays the foundation to use the VAE for a 360-degree sensor range, which an autonomous vehicle requires.

Validating the decoded VAE results in 3D is critical for assessing the integrity of the LiDAR point cloud. Therefore, a 3D projection of the encoded/decoded LiDAR depth map was performed, and the results are presented in the next section.

4.2.2 Qualitative performance assessment

To assess the VAE encoding and decoding performance qualitatively, visual inspections of the VAE output was performed in the 3D space. Therefore, the densified VAE output was filtered for valid pixels, projected into 3D as a point cloud, and visually compared to the reference (raw) point cloud. Figure 9 shows results specific to camera view 02 (left front camera) and compares the results of experiments 01 – 07. Figure 9A displays the error distribution curve. Additionally, Figure 9B compares the reference 3D LiDAR point cloud (red) with the 3D projected VAE

output (gray). For clarity, the camera view and 2D depth map of the driving scene are included in Figure 9B.

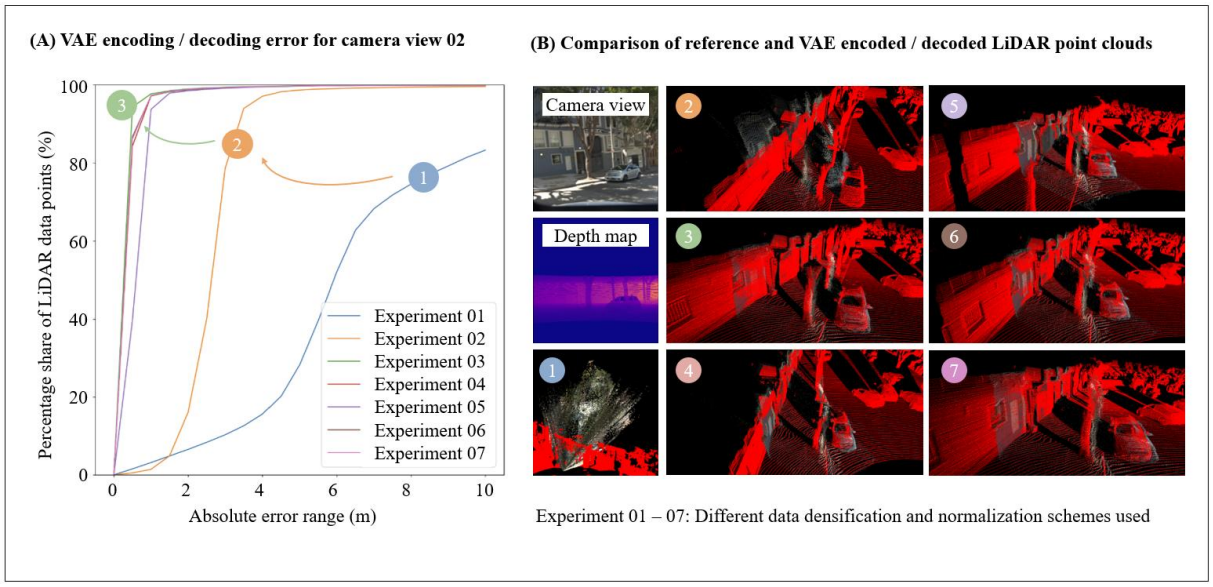


Figure 9: Developing domain specific data densification and data normalization approaches is required to achieve decent encoding / decoding performance with Variational Autoencoders on sparse 2D LiDAR depth maps. Significant performance improvements are achieved comparing experiment 01 with experiment 03. (A) Presents quantitative error metrics for the encoding / decoding of 2D LiDAR depth maps using an RGB pre-trained VAE: It can be seen what percentage of LiDAR data points fall within an error range which is calculated pointwise. (B) Compares the ground truth 3D LiDAR point clouds (red) with the decoded LiDAR point clouds (grey). It allows visual inspection in 3D and assessment of the geometric integrity of the encoded / decoded LiDAR point clouds.

Table 2Table 3 provides a detailed assessment of all seven experiments performed. Furthermore, the following key insights can be drawn:

1. **Good performance**: The best-performing models show consistency between the red reference LiDAR point cloud and the gray VAE encoded/decoded LiDAR point cloud (experiment 04). Based on this, it was concluded that using the RGB-trained VAE for the encoder/decoder in the latent stable diffusion model pipeline is a viable option.

2. **Data densification introduces a step change in VAE performance**: Data densification is required to close the offset between the reference and VAE encoded/decoded point clouds (comparing experiment 01 vs. experiment 02).

3. **Non-linear normalization is required to serve the full 150 m range**: Tanh-normalization is critical to overcome geometric distortions of objects when encoding/decoding the full 150 m LiDAR range. Comparing the results for experiment 02 with experiment 04 shows that better geometric integrity can be achieved using tanh-normalization schemes.

4. **Training the VAE specifically on LiDAR depth maps would be required to overcome remaining errors**: When analyzing experiments 04 – 07, one needs to note that geometric distortions and an offset between the decoded and the reference LiDAR point clouds, especially in the near range, remain. Consequently, using tanh-normalization with an appropriate scaling factor is an enabler, but it remains a limiting factor that the VAE is only trained on RGB data and not on LiDAR depth maps. Therefore, future work should also involve training of the VAE to reduce remaining geometric errors. Despite the fact that tanh-normalization yielded significantly better geometric integrity when working with LiDAR data across a 150 m range, it is also true that the performance of the tanh normalization is sensitive to the scaling factor and introduces additional complexity such as "wobbling of edges". Therefore, both normalization approaches were considered when training latent stable diffusion models capable of generating both camera images and LiDAR depth maps simultaneously. This would allow us to assess the performance on generative downstream tasks.

Table 3: Performance comparison, insights and conclusions about seven different data densifications and normalization approaches when encoding / decoding sparse LiDAR data with a RGB pre-trained Variational Autoencoder

#	**Normalization and densification approach**	**Observed VAE encoding/decoding performance and conclusions**
01	**No densification** **Linear normalization** Clipping to 0, 150 m	Only very poor encoding/decoding performance is observed as the data sparsity in LiDAR depth maps disrupts VAE convolutions.
02	**2x densification** **Linear normalization** 2x MaxPooling* Clipping to 0,150 m	Better VAE performance achieved. Consequently, smoothing is required to make VAE convolutions work. Yet, some sparse areas remain in the 2D LiDAR depth maps and encoding/decoding sharp edges remains a challenge.
03	**2x densification** **Linear normalization** MaxPooling* **Near range clipping** Clipping to 0, 30 m	Limiting the LiDAR depth range to 30 m significantly improves the VAE performance. Repurposing RGB pretrained VAEs for LiDAR depth maps works well for the near range (e.g., indoor scenes). It remains a challenge to precisely encode/decode the full dynamic range required for LiDAR depth captured by autonomous vehicles. To do so, normalization plays a critical role in improving accuracy, especially for the side cameras.
04	**2x densification** **Tanh normalization** Tanh factor: 0.03 2x MaxPooling* Clipping to 0, 150 m	Switching to non-linear linearization schemes can enable decent VAE performance across a larger LiDAR depth range (up to 150 m). Scaling the tanh-function accordingly to the data distribution is critical; otherwise, small depth values get mapped to 0, and large depth values get mapped to 1. To prevent this information loss, choosing a suitable tanh factor is very important to achieve good performance, especially for the front and back camera. Overall, better performance is achieved across all cameras compared to case 02. Yet the performance is worse compared to case 03.
05	**2x densification** **Tanh normalization** Tanh factor: 0.02 2x MaxPooling* Clipping to 0, 150 m	A tanh factor of 0.02 seems to be too small as the walls become wobbly and the offsets for the near-range cameras become severe. Significant deviations in the far range for the front camera can be observed between the reference LiDAR point cloud and the encoded/decoded LiDAR point cloud.
06	**2x densification** **Tanh normalization** Tanh factor: 0.03333 2x MaxPooling Clipping to 0, 150 m	Increasing the tanh factor to 0.03333 leads to a minimization of the encoding/decoding error across all cameras. Yet, the encoding/decoding performance for the front and back cameras still have weaker performance than side cameras often capturing the near-range cameras. The performance is worse than for case 3 which limits the data to the near range. Improvement is achieved with regards to the geometric integrity of the scene: The walls look straight, yet not as good as with linear normalization. Overall, some cameras are better, others are worse than for case 4.
07	**8x densification** **Tanh normalization** Tanh factor: 0.03333 8x MaxPooling* Clipping to 0, 150 m Avg. all channels	Two more data processing approaches are tested: 1) highly densifying the 2D depth maps (for eight iterations of max pooling to close remaining holes. 2) Averaging the three decoded channels. It would be expected that the additional densification benefits the VAE performance and that the averaging across the three channels leads to more stable results. Yet the improvements remain small or the observed error metrics become slightly worse.

4.3 Results and evaluation of trained diffusion models for driving scene generation

Table 4: Experimental setups for trained stable-diffusion models which generate multi-view camera images and corresponding LiDAR 2D depth maps

Model #	Description	# Poolings	Kernel size	Type of densification	Masking	Normalization	Max train steps	Training loss	Validation loss
Model 01	Unmasked, 8x max pool, tanh	8	3x3	Max pooling	No	Tanh s = 0.03	302.5k	0.3853441477	0.454198
Model 02	Masked, 2x max pooled, tanh	2	3x3	Max pooling	Yes*	Tanh s = 0.03	320.5k	0.3057099879	0.2910346
Model 03	Masked, 2x max pooled, lin.	2	3x3	Max pooling	Yes*	Linear max min (150 m)	383k	0.298086226	**0.2828812**
Model 04	Masked, 2x median pooled, tanh	2	3x3	Median pooling, zeros ignored	Yes*	Tanh s = 0.03	171.5k	0.3053328395	0.3088916
Model 05	Masked, 2x median pooled, lin.	2	3x3	Median pooling, zeros ignored	Yes*	Linear max min (150 m)	199k	**0.2980561554**	0.3006997
Model 06	Masked, 1x median pooled 5x5 kernel, tanh	1	5x5	Median pooling, zeros ignored	Yes*	Tanh s = 0.03	228k	0.3032646775	0.30616406
Model 07	Masked, 1x median pooled 5x5 kernel, lin.	1	5x5	Median pooling, zeros ignored	Yes*	Linear max min (150 m)	81.5k	0.3025480509	0.31738397

Training dataset size: 65941 record (180 T), test data set size: 22997 tf records (41T), total ~ 1 million records, minimum loss values highlighted in orange, * each LiDAR depth map was masked to the LiDAR field of view

Quantification of improvement in model training loss: Different latent stable diffusion models were trained using the same model architecture but with different input datasets. The preparation of the datasets included various strategies with regards to data masking and densification of the LiDAR data. For example, either max pooling or kNN median pooling with zero exclusion was applied as the densification algorithm, varying the number of iterations.

Table 4Table 2 summarizes the experimental setup for models 01 – 07. The lowest training loss was achieved for model 05, while model 03 achieved the lowest validation loss.

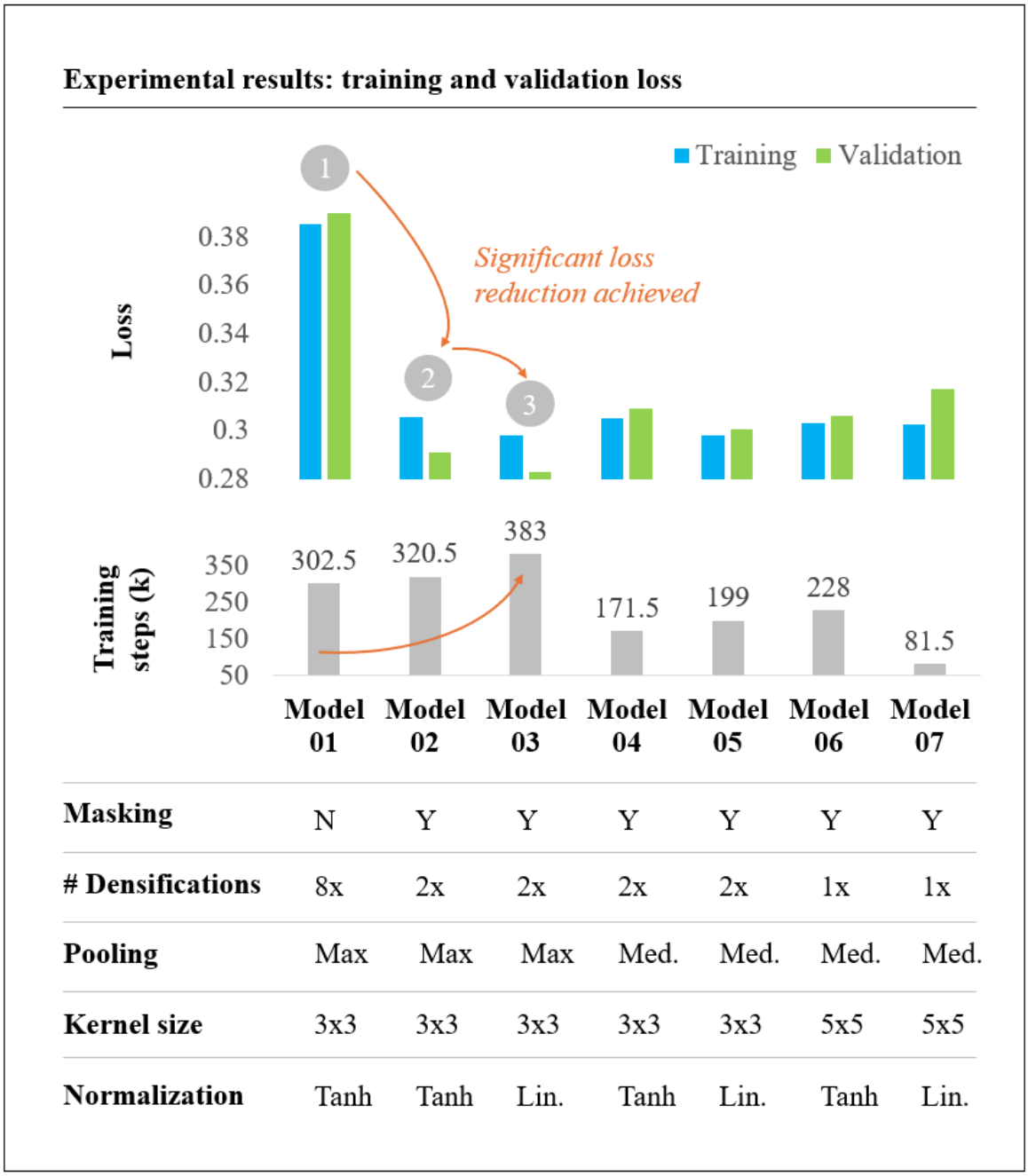


Figure 10: Training and validation loss for models 01 - 07

Comparing the training and validation loss across the seven different models leads to two key insights (see Figure 10):

1. By reducing the data densification iterations from eight to two and applying a mask corresponding to the LiDAR field of view to each LiDAR depth map, a significant reduction in training and validation loss can be achieved (from 0.385 to 0.306 in training loss). This becomes clear when comparing model 01 and model 02.

2. Increasing the number of training steps from approximately 320k to approximately 383k furthermore plays a role in achieving lower training and validation losses, thus increasing model performance. This trend becomes apparent when comparing model 02 and model 03.

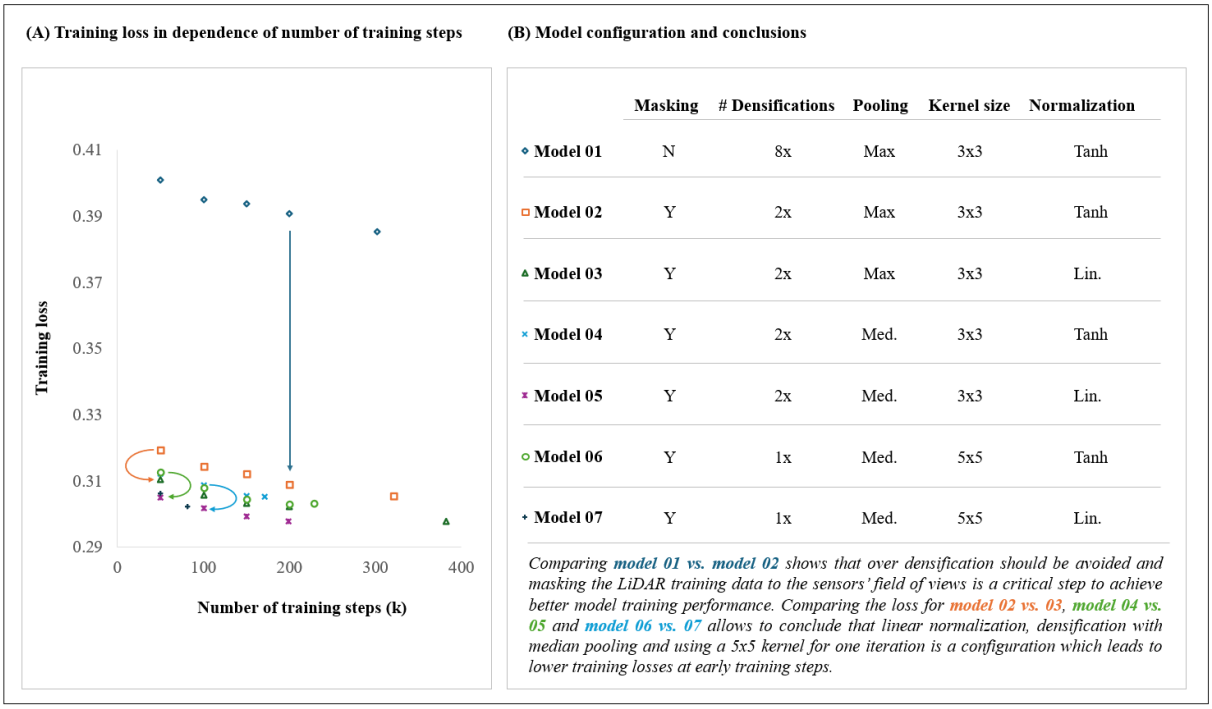


Figure 11: Comparison of the training loss achieved at different training steps for the trained latent stable diffusion models 01 – model 07

Further comparison of the training loss at an equal number of training steps across the seven different experiments allows us to draw conclusions about the applied normalization and data densification strategies. Figure 11 displays the following two additional trends:

1. **Linear normalization leads to lower training loss:** Models trained on data that was linearly normalized appear to achieve lower training loss values. Figure 11 illustrates

this trend by comparing model 02 vs. model 03, model 04 vs. model 05, and model 06 vs. model 07.

2. **kNN median pooling yields lower training loss:** Comparing model 03 (green) and model 05 (pink) allows us to conclude that improving the data densification technique by using kNN median pooling versus max pooling leads to a lower training loss and potentially better generative model performance. While model 03 has been trained for the largest number of training steps (383k), model 05 already achieved a lower training loss at 199k training steps. It is expected that training model 05 for more training steps would lead to an additional reduction in both training and validation losses. This is important to note when evaluating the data densification approaches. Of course, it would have been ideal to train all models for the same number of training steps, yet compute resources were limited and it was important to draw directional conclusions early on.

Figure 12 provides examples of generated camera views and the corresponding LiDAR depth maps for all seven latent stable diffusion models which were trained. Across all trained models, consistent alignment between the generated camera view and the corresponding LiDAR depth maps can be observed. This demonstrates successful multimodality, as both data domains—RGB camera images and 2D LiDAR depth maps—are generated consistently.

Additionally, the examples provided in Figure 12 show that the generated RGB images are geometrically consistent and display a diverse set of driving scenes. Model 07, only trained for ~81.5k training steps, shows lower quality in the generated RGB camera images compared to models 02 – 06. Therefore, it will be excluded from additional model comparisons in the subsequent section.

As concluded above, model 05 appears to be the most promising, given its low training and validation loss and the improved data densification strategy applied (relying on kNN median

pooling). The next section will therefore focus on generative results created when prompting model 05.

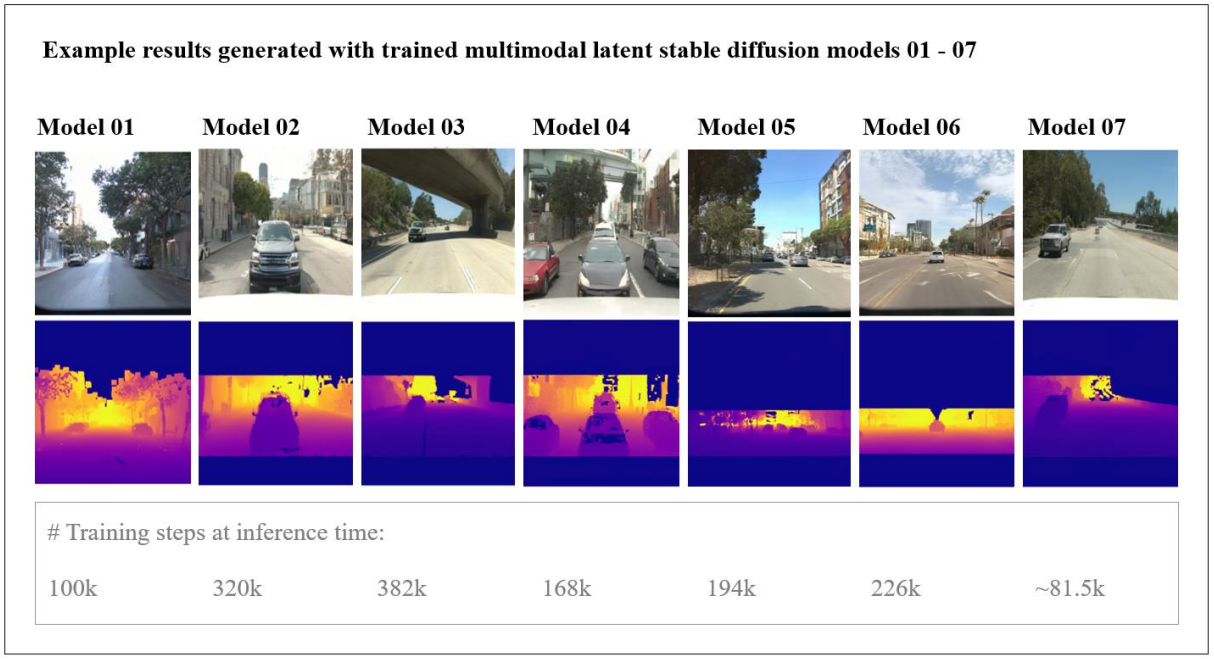


Figure 12: Examples of driving scenes generated with the trained latent stable diffusion models 01 – model 07. Each model generates camera views and the corresponding LiDAR depth maps. The number of training steps at inference time varied across the seven different models.

The goal of this research is to use the trained latent stable diffusion models for the simulation of sensor signals that the cameras or the LiDAR sensors of an autonomous vehicle would generate. Therefore, a two-step approach was invented: First, the trained diffusion model would generate eight camera views paired with the corresponding eight 2D LiDAR depth maps ((Figure 13A and Figure 13B). Second, the 2D LiDAR depth maps are projected into 3D space as a point cloud (Figure 13C). To do so, a camera model for which a standardized set of intrinsics and extrinsics were used. This novel approach presents an alternative to conventional sensor simulation techniques which rely on reconstruction vs. generation. It is impressive to see a high level of geometric integrity achieved even in the 3D space: The shape of the car looks decent, the road geometry looks realistic, and areas where no sensor signals are expected to be received (sky) also have no returns in the generated LiDAR driving scene. Of course, analyzing the driving scene closely also reveals room for improvement, especially around the edges of objects where LiDAR sensors often receive scattered signals. Additionally, it remains a

challenge for the trained diffusion models to generate sensor signals for plants, bushes, and trees consistently as these objects also have low reflectivity and tend to scatter LiDAR signals.

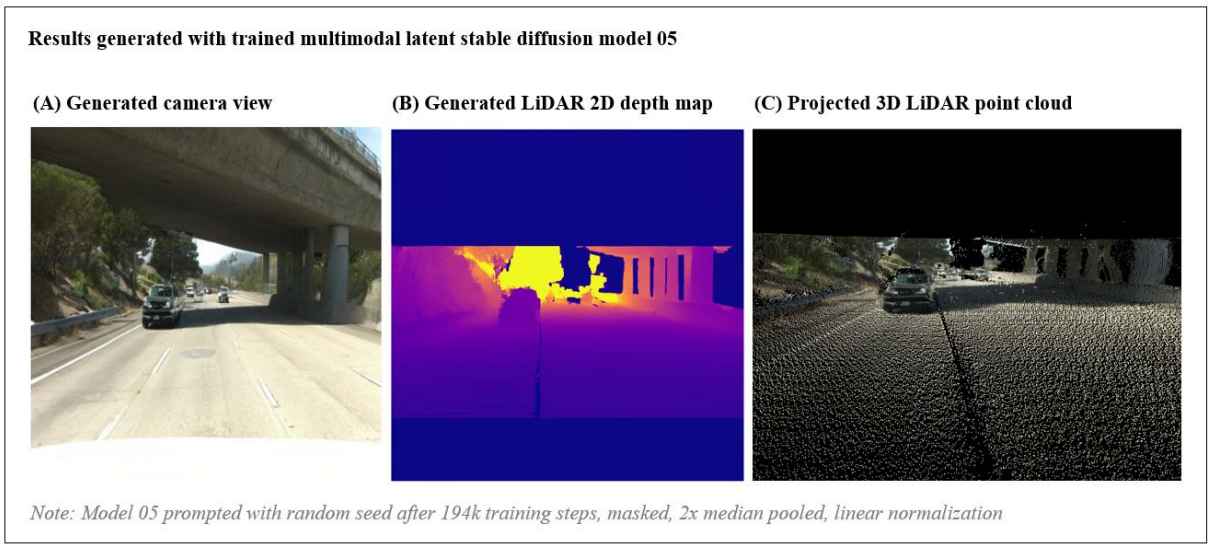


Figure 13: Selected view of a driving scene generated with model 05 after 194k training steps. The model is capable of simultaneously generating RGB camera views and 2D LIDAR depth maps. A camera model was used to project the 2D LiDAR depth map into the 3D space to receive a LiDAR point cloud which is representative of a sensor signal an autonomous vehicle measures.

The trained stable diffusion models have the novel capabilities of generating a 3D world model in the camera and LiDAR data domain. Figure 14 demonstrated the capability of generating panoramic consistent camera views (Figure 14A) and 2D LiDAR depth maps (Figure 14B). These views cover a full 360-degree range for both data modalities: cameras and LiDAR depth map. Additionally, a full 360-degree 3D LiDAR point cloud was constructed by projecting the 2D LiDAR depth maps into the 3D space using a camera model (Figure 14C).

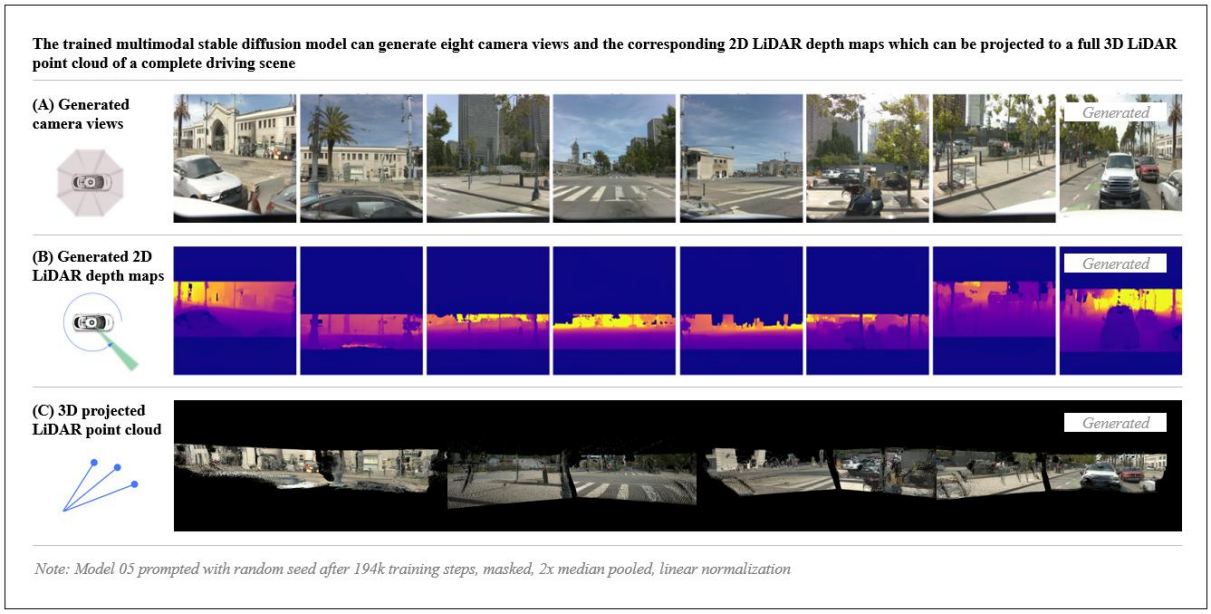


Figure 14: 360-degree driving scene generated with model 05. Eight camera views and eight LiDAR depth maps are generated simultaneously which span the full panoramic range an autonomous vehicle observes. The eight 2D LiDAR depth maps were projected into 3D and constitute a 360-degree LiDAR point cloud simulating the signal of a LiDAR sensor of an autonomous vehicle.

Given the diversity in driving scene conditions included in the training dataset, the trained stable diffusion model can also generate scenarios with different weather or lighting conditions. Figure 15 shows generated camera images and 2D LiDAR depth maps for rainy, foggy, or sunny conditions, which could occur at different hours of the day. Upon inspecting Figure 15, one can see that rainy conditions impact not only the camera image but also that the trained stable diffusion model is capable of generating LiDAR depth maps displaying an increased level of signal scattering caused by rain. Therefore, it can be demonstrated that for both data domains, realistic sensor signals specific to the faced weather conditions can be generated.

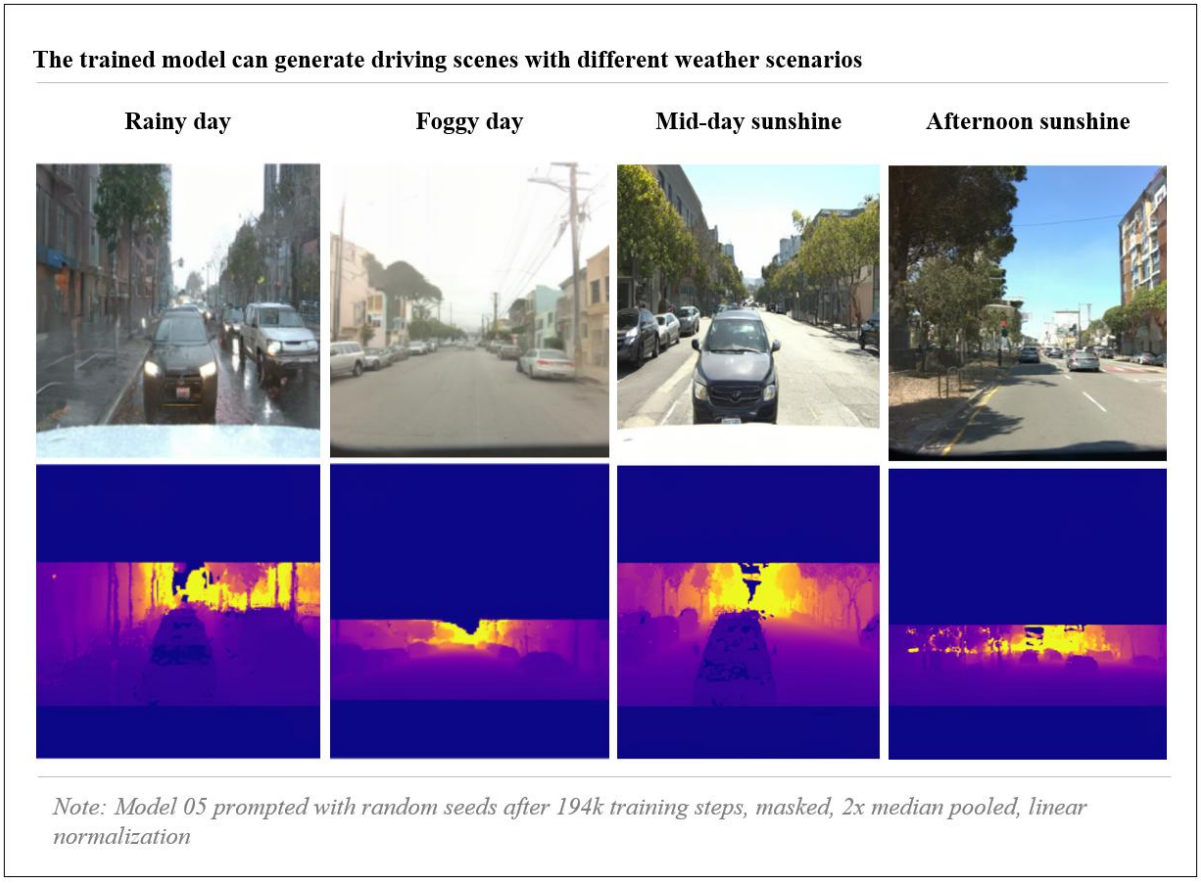
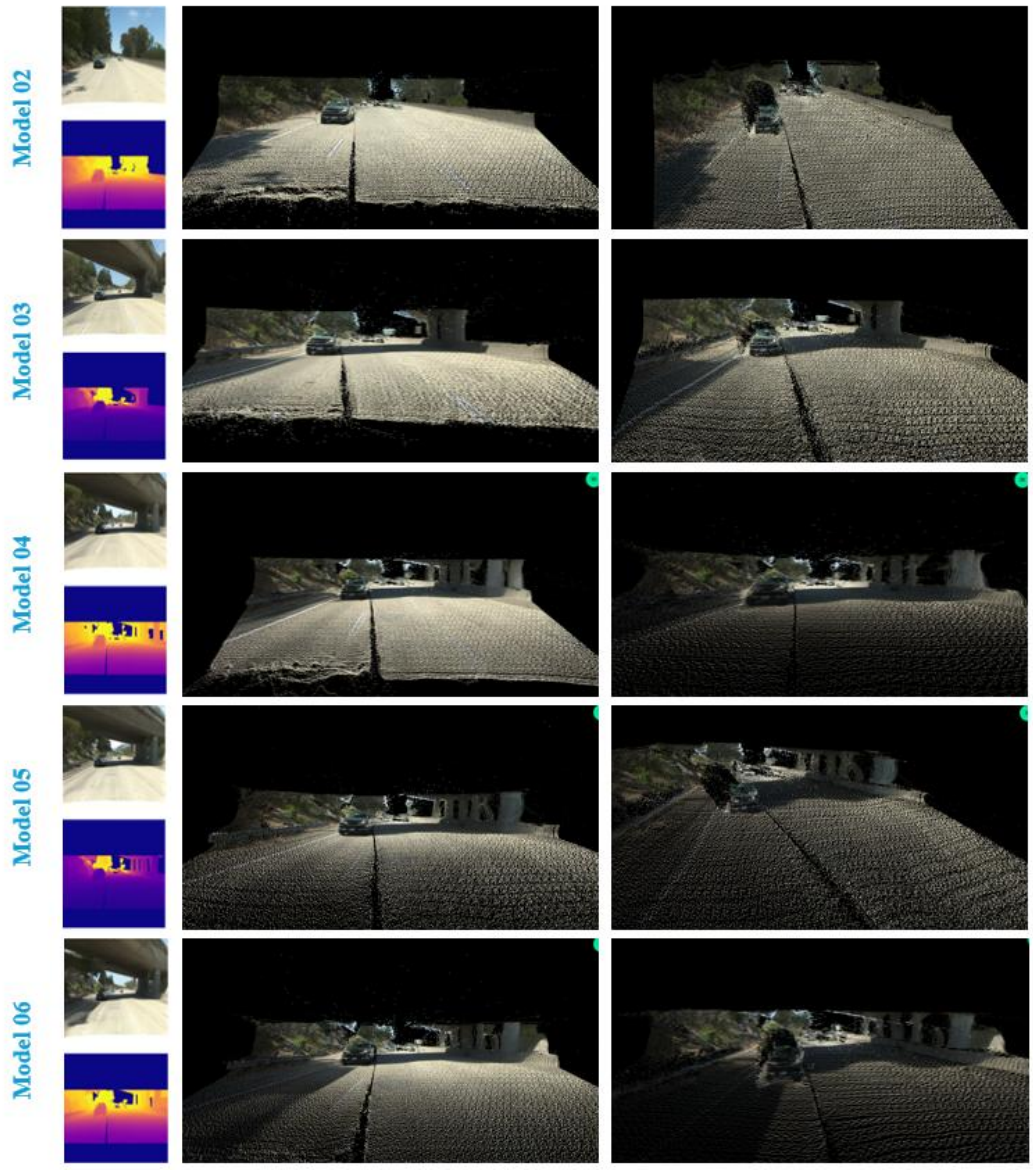


Figure 15: The trained latent stable diffusion model is capable of generating driving scenes with different weather and lighting conditions.

To assess the performance of the trained stable diffusion models, a stepwise approach is taken: First, analyzing the training and evaluation loss. Second, visually inspecting the generated results in 2D, focusing on the geometric integrity of objects, alignment between the camera views and LiDAR depth maps, panoramic capability to cover a full 360-degree range, and diversity of generated driving scenarios with regard to scene and weather conditions. As a third step, the generated LiDAR signals are assessed in 3D space. This is an important evaluation criterion since the goal of the research is to simulate sensor signals for both the camera and LiDAR sensors. The camera RGB images are 2D sensor signals, while, in contrast, the LiDAR sensors capture 3D point clouds of reflected laser signals.

Comparison of generated camera views, 2D LiDAR depth maps and 3D LiDAR point clouds for the trained latent stable diffusion models 02 - 06



	Model 02	Model 03	Model 04	Model 05	Model 06
Masking	Y	Y	Y	Y	Y
# Densifications	2x	2x	2x	2x	1x
Pooling	Max	Max	Med.	Med.	Med.
Kernel size	3x3	3x3	3x3	3x3	5x5
Normalization	Tanh	Lin.	Tanh	Lin.	Tanh
#Train steps	320k	382k	168k	194k	226k

Figure 16: Different perspectives of 3D LiDAR point clouds generated with trained multimodal latent stable diffusion models 02 - 06

Figure 16 compares generated camera views, 2D LiDAR depth maps, and two different perspectives of the projected 3D LiDAR point clouds. The results were obtained by prompting trained stable diffusion models 02 – 06 with the same random seed and selecting a rear camera view example. When analyzing the 3D point clouds, special attention was paid to:

1. Alignment between the camera view and LiDAR point cloud,
2. Geometric integrity of the generated car or the pillars of the bridges,
3. Scattering of the LiDAR signals on the edges of the car.

In general, very good alignment between the camera view and the 3D LiDAR point cloud is achieved for all five models. Small deviations can be observed at the edges of trees or bridges. LiDAR sensors are not expected to receive return values when pointed at the sky. Therefore, the blue points (sky) in the 3D LiDAR scene should have led to missing returns instead. Yet, in general, LiDAR sensors receive scattered signals at the edges of sharp objects or objects with low reflectivity. Therefore, one could expect that the additional points generated for the sky close to the edges of trees or bridge can be induced due to two aspects: the scattering nature of sensor signals captured in the training data and limitations in the trained stable diffusion model or the used Variational Autoencoder leading to offsets.

Model 02 generated a simple scene where no bridge is present. The generated vehicle shape looks good, while significant scattering of the LiDAR signal can be observed on the left side of the vehicle. For model 03, the generated pillars of the bridge have an expected shape which suggests static stability of the bridge. Both pillars are nicely vertically aligned. The generated vehicle shows good geometric integrity. Some scattering of LiDAR signals can be observed on the left side of the vehicle, which appears to be less prevalent compared to model 02.

Model 04 generated a bridge with more nuanced details – the bridge has more pillars. Their shape appears to be vertical, only slight distortions can be observed on the right edge of the scene. Like model 03, some LiDAR signal scattering can be observed on the left edge of the vehicle, which overall has good geometric integrity. Model 04 overall generated very promising results. When prompted, model 05 generated a comprehensive driving scene which included a bridge with multiple pillars, a main car, and additional cars in the background. The pillars of the bridge appear bent in the 3D space, leaving a realism gap in the driving scene. Yet, the generated LiDAR signal appears to have sharper edges and less edge scattering, which is desirable. Inspecting the point cloud from different perspectives and zoom settings was performed as an inspection method.

Model 05 appears to be superior to model 06. As model 06 generated a scene with more edge scattering present on the left side of the vehicle and the geometric integrity of the bridge leaves room for improvement.

In conclusion, the first-of-its-kind generative multi-modal latent stable diffusion model was researched and developed, which is capable of generating 360-degree camera views, 2D LiDAR depth maps, and 360-degree 3D LiDAR point clouds for driving scenarios. This invention lays the foundation for a new approach to simulating 3D worlds in which autonomous vehicles can be tested and validated. It has the potential to complement or replace conventional technologies that rely on reconstructive methodologies.

5 Discussion

5.1 Significance of findings

The developed generative multi-modal latent stable diffusion models lay the foundation for transformative advancements in the field of autonomous vehicle simulation technologies. The developed model architecture uniquely supports the generation of complete 360-degree camera views along with 2D LiDAR depth maps and 3D LiDAR point clouds. This multimodal approach does not only allow the simultaneous and perspective-consistent generation of synthetic camera and LiDAR sensor signals, but it also allows great flexibility in the diversity of newly generated driving scenarios. The demonstrated capabilities extend beyond those of traditional simulation techniques, which are predominantly based on reconstructive methodologies and simulate traffic objects [23] rather than sensor signals. Accelerating the time to generate new scenarios of high detail and great diversity in driving scenes is of significant value to the autonomous driving industry since it could allow to accelerate the simulation, testing and validation operations. [24]

By enabling the simulation of realistic driving scenarios from any conceivable angle, the model provides an unprecedented level of panoramic coverage of simulated sensor signals. This comprehensive coverage is crucial, as it can correspond to the full detection range of sensors used in autonomous vehicles.

Overall, the impact of this work lies in demonstrating that it is possible to leverage latent stable diffusion models for the realistic simulation of synthetic camera and LiDAR data signals. The findings lay the groundwork for novel approaches to simulating driving scenarios, enhancing virtual testing and validation capabilities for the autonomous driving industry. This could in turn lead to a reduction in the required on-road test and validation milage which would not only lead to cost savings, but potentially also accelerate the time to expand to new geographies.

5.2 Comparison with existing methodologies

Traditionally, vehicle simulation technologies have relied heavily on reconstructive methodologies, which recreate driving scenarios based on real-world data collected through sensors and cameras. [24] Critical scenarios are rare in real-world driving, such that many miles need to be collected to capture these, therefore comprehensive on-road testing is resource intense. [23,24] While reconstructive methods are effective for replicating known conditions, they are inherently limited by the diversity of the data collected, often confining testing to a narrow set of scenarios. [24] This limitation is significant, particularly as autonomous vehicles must operate reliably under a wide variety of unpredictable conditions not always present in the initial data sets. [23]

In stark contrast, the trained stable diffusion model has the potential to fabricate an almost infinite array of driving conditions and scenarios. The trained model does not just mimic real-world sensor data in a realistic way but can be used to generate new, unrecorded scenarios, thereby greatly expanding the range of possible test cases. The ability to generate these diverse scenarios can accelerate testing and development cycles, and decrease the time and cost associated with scaling autonomous vehicles to expanded areas.

5.3 Limitations of the study

Despite the achieved advancements, the generative multi-modal latent stable diffusion model does present some areas that warrant further research and refinement. One notable challenge is the scattering of LiDAR signals, especially prominent around the edges of objects and on surfaces that are naturally less reflective. Real-world LiDAR signals are likely to be scattered around the edges of sharp objects, but amplified scattering effect can reduce the realism of simulated environment. This would potentially lead to slight inaccuracies in depth perception that could subtly influence an autonomous system's ability to accurately interpret and navigate

its surroundings. Such remaining gaps in simulation realism would limit their use for virtual testing of autonomous vehicles.

Additionally, the model sometimes produces geometric variances that could affect the realism of the simulated environments. These variances could appear as bent pillars, skewed building facades, or distorted road edges that do not conform to typical architectural and physical norms.

These geometric variances would need to be addressed in order to enhance the simulation realism and increase its viability to be used as virtual testbed for the autonomous driving industry.

Addressing these challenges involves refining the resolution and fidelity of the generated simulations. Future efforts might include integrating advanced physical rendering techniques to more accurately model the interaction of laser rays send by LiDAR sensor with various surfaces, enhancing the training algorithms to minimize the influence of minor anomalies in the training data, or employing more sophisticated transformer model architectures that can better generalize from the available data while minimizing the generation of unrealistic features.

By concentrating on these areas, the model can be enhanced to offer even more accurate and reliable simulations, narrowing the gap between virtual testing environments and real-world conditions.

Moreover, this study demonstrates novel baseline capabilities, yet additional benchmarking experiments including user studies and model evaluation on downstream task would be beneficial to more comprehensively evaluate model performance. This would be helpful to increase model understanding, derive improvement levers, and continue to increase the model performance and quality of generated sensor signals of novel driving scenarios.

6 Outlook & recommendation

This research addresses a critical challenge in the development and validation of autonomous vehicles: the reliance on extensive real-world data collection for performance evaluation and safety assurance. While simulation-based validation offers a promising alternative, traditional simulation technologies often suffer from a realism gap, limiting their effectiveness in replicating diverse and complex driving scenarios.

By leveraging advancements in generative artificial intelligence, this work introduces a novel **multimodal, multi-view latent stable diffusion model** capable of generating **eight camera views and corresponding LiDAR depth maps** for a full 360-degree perception of an autonomous vehicle. Unlike existing generative models that focus primarily on visual data, this framework integrates depth information with high geometric integrity, thereby enhancing the realism and utility of synthetic driving scenes.

Key contributions of this research include:

1. **Densification strategies for LiDAR data** using Variational Autoencoders (VAEs) to effectively encode and decode sparse depth maps.

2. **A multimodal generative model** that produces 360-degree camera views and LiDAR depth maps simultaneously, maintaining consistent perspectives and object geometry.

3. **A camera model for LiDAR-to-3D projection**, enabling the simulation of both camera and LiDAR sensor signals in their native formats.

This research lays a strong foundation for advancing synthetic scene generation for autonomous vehicle simulation and testing. Future work could explore the following areas:

Enhancing generative capabilities: Building upon the demonstrated capabilities of the generative multi-modal latent stable diffusion model, a focused approach should be taken to refine the precision of the model's output. This could involve introducing conditioning mechanisms that guide the generative process. For example, conditioning on weather conditions, road graphs, or specific scene descriptions improve the generative performance of the stable diffusion model and enable its user to more precisely control the generative process steering it towards the desired scenario.

Introducing a temporal component: This research has laid the foundations for generating the camera and LiDAR sensor signals of static driving scenarios. Future work could further build upon the rapidly advancing video generation methods and aim at the generation of dynamic driving scenarios. This could path the way to closed-loop simulation approaches and advanced simulation capabilities for the autonomous driving industry.

Complementing reconstructive techniques: To maximize the impact of generative simulation technologies, integrating them with existing reconstructive simulation methodologies presents a promising pathway to overcome remaining shortcomings and accelerate their adoption. By leveraging the strengths of both approaches—generative models for their flexibility and creativity in scenario generation, and reconstructive models for their grounding in real-world data—the combined framework could offer a more comprehensive toolset for scenario developers. This synergy could lead to more robust simulations.

Training the VAE on LiDAR data: A significant opportunity lies in optimizing the Variational Autoencoder (VAE) by specifically training it on LiDAR data collected directly from autonomous vehicles. Repurposing a VAE which is trained on RGB camera images is a viable option, yet it also limits the performance and geometric integrity of the LiDAR point cloud generated. Therefore, training the VAE on LiDAR depth maps, is desirable to further

improve the performance of the generative diffusion model, achieve higher geometric integrity and more accurate depth sensor signals in simulated environments.

Reducing diffusion steps for scalability: To enable the broader adoption and practical application of stable diffusion technologies, it is crucial to address the computational demands of the process, particularly the number of diffusion steps required. Reducing these steps without compromising the quality of the output could be an essential enabler for speeding up inference times. Faster inference times will make stable diffusion a more viable technique at scale, opening up possibilities for more frequent and extensive use in real-time simulation scenarios.

Adapting to alternative hardware configurations: Introducing the flexibility to scale the trained latent stable diffusion models to alternative hardware configurations which, e.g., have less cameras or do now comprise of a LiDAR system could be an additional research direction to pursue. This would allow not only to simulate sensor signals of critical driving scenes for one car model and sensor configuration but also enable the generation of driving scenes for varying car models or sensor configurations.

Conclusion

By addressing these focused areas, the potential of generative multi-modal, multi-view latent stable diffusion models in autonomous vehicle simulation can be realized. The identified future research directions align with current technological advancements and could pave the way for next-generation simulation tools. This research has the potential to enable more efficient virtual testing and validation, reducing the need for extensive on-road data collection while lowering operational costs and required mileage. In turn, this approach could accelerate deployment timelines and support scalability to new geographic regions. By generating high-fidelity synthetic driving scenes, the proposed framework enhances autonomous vehicle testing, ultimately contributing to safer and more efficient development of autonomous driving systems**.**

7 Acknowledgments

This work would not have been possible without the invaluable guidance and support of my academic advisors: Prof. Yoon Kim, and Prof. Rama Ramakrishnan. Their insightful feedback, expertise, and encouragement have played a crucial role in shaping the direction and quality of this research.

Sincere appreciation goes to my mentors, Atul Kumar, Max Jiang, Vincent Casser, and Bo Sun, for their wisdom, patience, and motivation. Their guidance has been instrumental in my academic and professional growth, providing clarity and inspiration throughout this journey.

The MIT Leaders for Global Operations (LGO) program office deserves special recognition for its continuous support. The program's commitment to interdisciplinary learning and leadership development has provided invaluable opportunities that have enriched this research.

Gratitude is also extended to all those who have supported this work, whether through discussions, technical assistance, or personal encouragement. Your contributions and belief in this research have been truly meaningful.

Allow me to say – thank you.

8 References

1 Gao, R. *et al.* Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314* (2024).

2 Ke, B. *et al.* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 9492-9502.

3 Haghighi, H., Samadi, A., Dianati, M., Donzella, V. & Debattista, K. Taming Transformers for Realistic Lidar Point Cloud Generation. *arXiv preprint arXiv:2404.05505* (2024).

4 Manivasagam, S., Bârsan, I. A., Wang, J., Yang, Z. & Urtasun, R. in *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 8272-8282.

5 Lu, J., Wong, K., Zhang, C., Suo, S. & Urtasun, R. in *2024 IEEE International Conference on Robotics and Automation (ICRA).* 16908-16914 (IEEE).

6 Hess, G., Tonderski, A., Petersson, C., Åström, K. & Svensson, L. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 7438-7447.

7 Hu, Q., Zhang, Z. & Hu, W. RangeLDM: Fast Realistic LiDAR Point Cloud Generation. *arXiv preprint arXiv:2403.10094* (2024).

8 Nakashima, K. & Kurazume, R. in *2024 IEEE International Conference on Robotics and Automation (ICRA).* 14724-14731 (IEEE).

9 Mildenhall, B. *et al.* Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**, 99-106 (2021).

10 Yu, Z. *et al.* SGD: Street View Synthesis with Gaussian Splatting and Diffusion Prior. *arXiv preprint arXiv:2403.20079* (2024).

11 Gu, J. *et al.* in *International Conference on Machine Learning.* 11808-11826 (PMLR).

12 Kim, S. W. *et al.* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 8496-8506.

13 Tao, T. *et al.* in *Proceedings of the 32nd ACM International Conference on Multimedia.* 390-398.

14 Zyrianov, V., Che, H., Liu, Z. & Wang, S. LidarDM: Generative LiDAR Simulation in a Generated World. *arXiv preprint arXiv:2404.02903* (2024).

15 Ran, H., Guizilini, V. & Wang, Y. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 14738-14748.

16 Ho, J. *et al.* Video diffusion models. *Adv. Neural Inf. Process. Syst.* **35**, 8633-8646 (2022).

17 Deng, B. *et al.* in *ACM SIGGRAPH 2024 Conference Papers.* 1-11.

18 Nunes, L., Marcuzzi, R., Mersch, B., Behley, J. & Stachniss, C. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 14770-14780.

19 Wang, Y. *et al.* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 14749-14759.

20 Giuliano, A., Gadsden, S. A., Hilal, W. & Yawney, J. in *Sensors and Systems for Space Applications XVII.* 124-134 (SPIE).

21 Bond-Taylor, S., Leach, A., Long, Y. & Willcocks, C. G. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence* **44**, 7327-7347 (2021).

22 Pinheiro Cinelli, L., Araújo Marins, M., Barros da Silva, E. A. & Lima Netto, S. in *Variational methods for machine learning with applications to deep networks* 111-149 (Springer, 2021).

23 Wang, Z., Ma, J. & Lai, E. M.-K. A Survey of Scenario Generation for Automated Vehicle Testing and Validation. *Future Internet* **16**, 480 (2024).

24 Zhou, J., Wang, L., Meng, Q. & Wang, X. DiffRoad: Realistic and Diverse Road Scenario Generation for Autonomous Vehicle Testing. *arXiv preprint arXiv:2411.09451* (2024).

25 Generative AI was used to review and correct the grammar and language to improve readability and clarity

9 List of tables