

Journal Pre-proof

HuGe: Towards **H**uman-controllable image **G**eneration in autonomous driving

Yuanzhi Zeng, Shiwei Chen, Yutian Zhang, Dong Sun, Yong Wang,
Haipeng Zeng



PII: S2468-502X(25)00045-2

DOI: <https://doi.org/10.1016/j.visinf.2025.100262>

Reference: VISINF 100262

To appear in: *Visual Informatics*

Received date: 17 December 2024

Revised date: 28 April 2025

Accepted date: 4 August 2025

Please cite this article as: Y. Zeng, S. Chen, Y. Zhang et al., *HuGe*: Towards **H**uman-controllable image **G**eneration in autonomous driving. *Visual Informatics* (2025), doi: <https://doi.org/10.1016/j.visinf.2025.100262>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2025 The Authors. Published by Elsevier B.V. on behalf of Zhejiang University and Zhejiang University Press Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

HuGe: Towards Human-Controllable Image Generation in Autonomous Driving

Yuanzhi Zeng^a, Shiwei Chen^a, Yutian Zhang^a, Dong Sun^b, Yong Wang^c and Haipeng Zeng^{a,*}

^aSchool of Intelligent Systems Engineering, Sun Yat-Sen University, Shenzhen, China

^bAlgorithm Department, CARIZON, Shanghai, China

^cCollege of Computing and Data Science, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:

Visualization
Autonomous Driving
Interactive Image Generation
Generative Model

ABSTRACT

The rapid advancement of autonomous driving technology has reshaped the automotive industry, highlighting the need for diverse and high-quality image data. Existing image datasets for training and improving autonomous driving technologies lack rare scenarios like extreme weather, limiting the effectiveness and reliability of autonomous driving technologies. One possible way of expanding the dataset coverage is to augment the existing dataset with artificial ones, which, however, still suffers from various challenges like limited controllability and unclear corner case boundaries. To address these challenges, we design and develop an interactive visual analysis system, *HuGe*, to achieve efficient and semi-automatic controllable image generation. *HuGe* incorporates weather transformation models and a novel semi-automatic knowledge-based controllable object insertion method which leverages the controllability of convex optimization and the variability of diffusion models. We formulate the design requirements, propose an effective framework, and design four coordinated views to support controllable image generation, multidimensional dataset analysis, and evaluation of the generated samples. Two case studies, a metric-based evaluation and interviews with domain experts demonstrate the practicality and effectiveness of *HuGe* in controllable image generation for autonomous driving.

1. Introduction

In recent years, the rapid evolution of autonomous driving technology has significantly influenced the automotive industry, driving advancements in vehicle automation, safety features, and intelligent transportation systems. The performance and reliability of current autonomous driving technologies highly depend on the training datasets. Currently, most autonomous driving systems primarily utilize data collected from daily driving scenarios for training and evaluation [29, 35]. While the lack of sufficient diversity is a common challenge across various computer vision applications, it poses unique and critical challenges in the context of autonomous driving [51, 9, 15]. Specifically, autonomous driving systems must be capable of navigating highly dynamic and unpredictable real-world environments, which include rare but potentially hazardous scenarios such as extreme weather conditions (e.g., heavy snow, dense fog, or intense rain), unexpected road obstacles (e.g., fallen trees or debris), and unusual behaviors of other road users (e.g., jay-walking pedestrians or animals crossing the road). However, existing datasets often lack sufficient representation of these edge-case scenarios, which can significantly compromise the system's ability to generalize and perform safely under

such conditions [34]. In other words, the further improvement of model performance is greatly constrained by the data distribution of corner cases. When these corner cases occur, they can lead to inaccurate predictions by the model integrated into the autonomous driving systems, potentially compromising the system's safety and reliability in real-world scenarios. Although some studies specifically focus on extreme weather datasets for autonomous driving [25, 41], featuring a variety of characteristics for different weather conditions, they still lack effective methods to analyze and address imbalances in certain dimensions of the dataset.

Therefore, to enhance the safety and reliability of autonomous vehicles, it is crucial to ensure that the datasets can cover as many scenarios a driver can face in real-world situations as possible.

Expanding the coverage of special scenarios in datasets commonly involves gathering more data from the real-world and generating data for specific scenarios [43]. For the former, collecting datasets for autonomous driving often requires road testing [51], which, however, road testing is often expensive and difficult to cover a wide range of driving scenarios, such as various weather and road conditions. Additionally, some research uses driving simulation software to obtain virtual image samples that mimic real-world scenarios [32]. While these simulated data have been shown in some studies to supplement relevant datasets, there are concerns about the fidelity of the simulations, and whether these simulated driving scenarios can truly reflect real-world conditions [28]. The gap between simulation and reality underscores the limitations of relying on virtual environments for comprehensive dataset enhancement. Therefore, it is necessary to develop methods for generating

*Haipeng Zeng

 zengyzh25@mail2.sysu.edu.cn (Y. Zeng);

chenshw39@mail2.sysu.edu.cn (S. Chen); zhangyt85@mail2.sysu.edu.cn (Y. Zhang); sundongcandy@gmail.com (D. Sun); yong-wang@ntu.edu.sg (Y. Wang); zenghp5@mail.sysu.edu.cn (H. Zeng)

ORCID(s): 0000-0002-0339-0361 (H. Zeng)

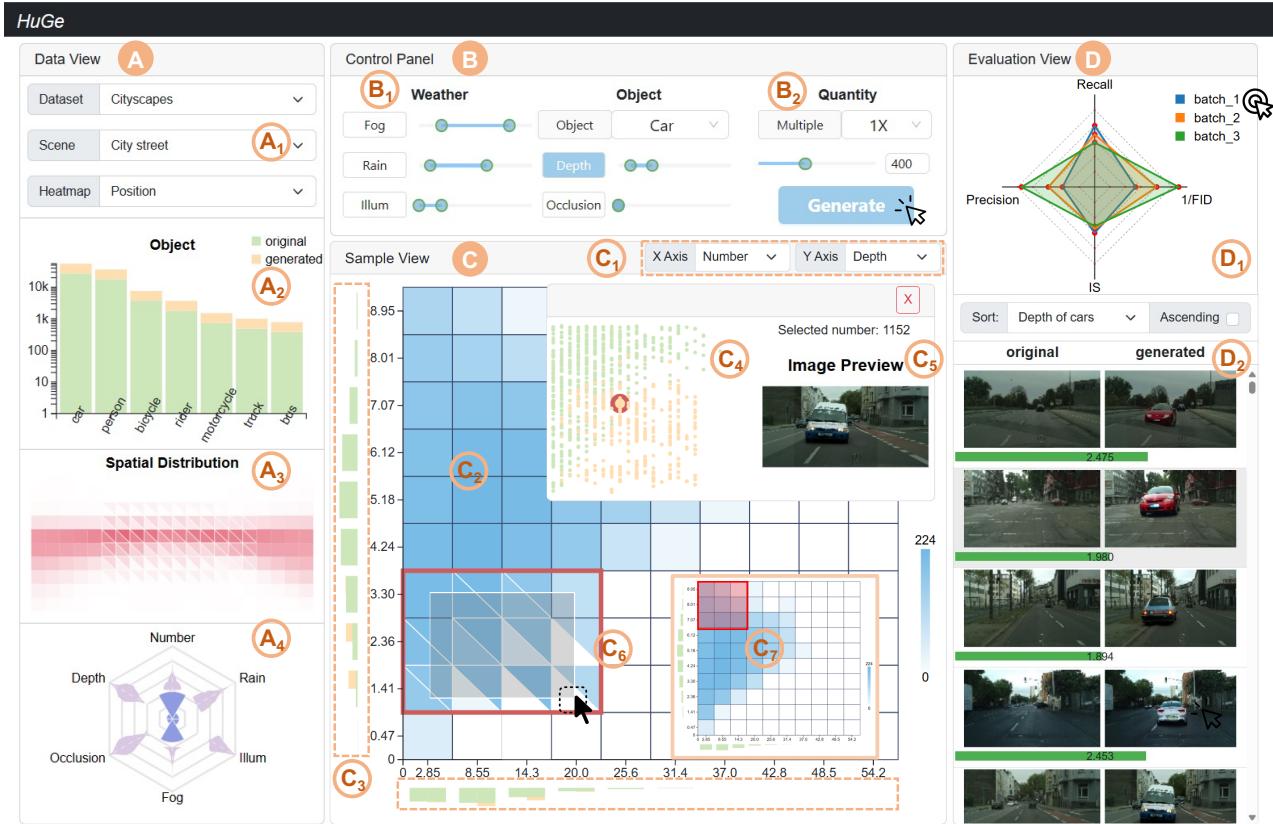


Figure 1: *HuGe* supports the generation of image samples for autonomous driving: The data view (A) shows an overview of a dataset, including the number of objects, the spatial distribution of objects and the distribution of different dimensions. The control panel (B) allows users to set the option of the generation process and generate new samples. The sample view (C) provides the distribution of the dataset with more details. The evaluation view (D) shows the generated results, where users can evaluate and compare different generated samples.

artificial data based on real images, introducing additional controllable factors. On the one hand, real-world image datasets themselves contain a wealth of information about real-world scenarios, aiding in training autonomous driving systems to better understand and adapt to the complexity and unpredictability of the real world. On the other hand, incorporating controllable factors into existing image data helps developers design specific scene images they need at a lower cost.

However, it is not easy to generate artificial data based on real images. Based on surveys of past research and expert interviews, we have identified three key challenges: **(1) Limited controllability.** The current methods for generating image samples offer limited controllability, particularly in their ability to customize image attributes across multiple aspects, such as lighting conditions, object positions and weather variations. This limitation highlights the absence of a unified framework—a comprehensive system capable of integrating these diverse aspects into a cohesive generation process. Such a framework would enable users to specify and control multiple attributes simultaneously, ensuring that the

generated images align more closely with specific requirements. **(2) Unclear boundary of corner cases.** Although existing studies have categorized corner cases for autonomous driving into different levels, they have not clearly defined and analyzed the boundaries of these corner cases, which hinders the generation of corner cases. **(3) Complex distribution of the dataset.** It is hard to know in which direction should we generate new images and how the generated images change the distribution, as well as whether they make up for the missing corner cases in the original dataset.

To address the aforementioned challenges, we designed and developed *HuGe* (Fig. 1), an interactive visual analysis system tailored for the artificial generation of image samples for autonomous driving. Over approximately two months, we engaged in close collaboration with experts in autonomous driving to define the system's design requirements and ensure its alignment with real-world application needs. Based on the derived tasks, a complete pipeline and four coordinated views are designed to support controllable image generation. To improve the controllability of artificial image generation, we identify a series of controllable factors (e.g., weather, illumination, object positioning, and degree of occlusion), and design a highly efficient, semi-automatic,

and controllable sample generation method. This method employs a hybrid methodology that integrates convex optimization and Conditional Variational Autoencoders (CVAE) to identify candidate locations for obstacles while leveraging a diffusion model to generate the obstacles in located position. To explore the boundary of corner cases, *HuGe* allows users to adjust parameters of controllable factors and generate different images. Further, to easily explore the complex change of the dataset, *HuGe* supports multidimensional analysis and mining of datasets, as well as evaluation of the generated samples visually. We demonstrate the effectiveness of our method through two case studies covering object insertion and weather condition adjustment, a metric-based evaluation, as well as interviews with domain experts. In summary, the primary contributions of this paper are:

- We develop *HuGe*, an interactive visual analytics system for controllable image generation and exploration in autonomous driving scenarios. *HuGe* supports multidimensional analysis, dataset mining, and evaluation of generated samples.
- We provide an effective paradigm for controllable image generation that considers weather transformation and object insertion, making it work for a variety of driving scenarios.
- We introduce a semi-automatic knowledge-based object insertion method that combines traditional algorithms with generative models, allowing users to efficiently expand dataset coverage based on specific needs..
- We conduct two case studies, a metric-based evaluation, and interviews with domain experts, providing comprehensive evidence of the usefulness and effectiveness of *HuGe* for controllable image generation in autonomous driving.

2. Related Work

This section presents studies related to our research in three categories, namely, autonomous driving scenario image generation, controllable generation, and visualization exploration of image datasets.

2.1. Autonomous driving scenario image generation

Due to the difficulty in acquiring real autonomous driving scenario data, many researchers have turned to artificially generating data to synthesize new datasets [50, 53, 44, 37]. Some researchers use simulation data [2, 47], but the data produced by simulation platforms often lack reality and are dependent on the scenario designer's understanding of the environment. In contrast, datasets generated from data collected in real scenarios exhibit greater diversity and complexity [28, 8], which aids in training models to better generalize and handle a variety of different situations and scenarios. Since the dataset is directly sourced from practical

application scenarios, it is closely related to actual problems and applications. This enhances the practicality of the data, making models trained from the dataset more adaptable to real-world conditions.

In the context of autonomous driving scenario images, common generation methods include adversarial attack-based methods and knowledge-based methods. For adversarial attack-based methods, some researchers added specifically modified black dots, rectangles, or noise [37, 42, 48] to create adversarial attacks aimed at identifying and generating risky scenarios, but such adversarially modified data are almost impossible to exist in reality. With the advent and development of generative models, many studies utilized these models for transformations that are more aligned with real-world scenarios. For instance, some studies [53, 36, 45] used GANs to generate driving scene data under various adverse weather conditions. Liu et al. [32] simulated natural raindrops to perform adversarial attacks on traffic sign detection. However, these methods are limited as they only consider changes in dimensions of weather or illumination. In our work, we consider transformations across multiple dimensions including weather, illumination, object positioning, and degree of occlusion.

In knowledge-based methods, generation is primarily driven by incorporating expert experience or integrating external knowledge. For instance, Xu et al. [50] utilized GANs to generate day-night images and sharp cut-in scenarios. Deng et al. [11] generated various driving scene tests under different conditions based on human-written rules. While these methods have expanded the range of testable driving conditions and improved scenario coverage, they still exhibit limitations in generating more complex, real-world driving scenarios that encompass unpredictable and highly dynamic elements.

Our method integrates convex optimization with CVAE model [42], enabling automatic identification of appropriate locations for placing objects. Additionally, given the strengths and limitations of diffusion models, we employ a hybrid approach that combines diffusion models with rule-based methods for open-set generation. This allows users to quickly generate batches of realistic and complex scenes, catering to a broader range of scenarios than previously possible with traditional methods.

2.2. Controllable generation

In the fields of Natural Language Processing (NLP) and Computer Vision (CV), controllable generation is a pivotal research topic aimed at precisely manipulating the output of generative models. This concept focuses on generating data that meets specific requirements or follows predefined guiding conditions, such as text [21], images [27], or videos [20]. There are various methods to achieve controllable generation, mainly categorized into three types: quantitative control, qualitative control, and positional control.

In terms of quantitative control, methods like Styleflow [1] utilize sliders to precisely adjust specific attributes such as age or gender. For qualitative control, models like

OpenAI's DALL-E-2 [39] allow users to guide the image generation process by inputting text. In the context of qualitative control for autonomous driving, GAIA-1 [19] generates videos by taking input in the form of actions, text, and video. Another example is from a study [11], where users can customize traffic rules to control image generation. As for positional control, ControlNet [52] guides the image generation process of stable diffusion by using additional information such as depth maps or keypoints. Additionally, EditGAN and pix2pix [22, 31] allow adjustments to generated images by manipulating masks, though in some cases, these adjustments may still require user intervention depending on the complexity of the desired modifications.

To further streamline this process and minimize manual effort when significant modifications to image datasets are required, we propose an alternative approach. Instead of requiring users to modify individual images, our method enables users to visualize the dataset distribution and select regions of interest through bounding boxes. This approach provides an efficient way to modify the dataset, saving time and reducing human effort.

2.3. Visualization exploration of image datasets

In recent years, many research organizations have constructed a large number of representative large-scale datasets, which have significantly propelled breakthroughs in artificial intelligence across multiple research fields. However, with the continuous expansion of these datasets, traditional manual exploration and evaluation methods have increasingly revealed their limitations in terms of time consumption and inefficiency. Consequently, many researchers have constructed visual analysis systems from a human-computer interaction perspective. They utilize visualization tools to explore image datasets, aiming to understand these complex and large-scale data collections.

Chen et al. [7] developed a general model evaluation system applicable to major tasks in the field of computer vision, employing a variety of visualization methods such as matrix, table, and grid. This system can identify issues affecting model evaluation in individual image samples. Addressing the problem that common methods for exploring datasets are not applicable to large-scale datasets, Bertucci et al. [4] utilized treemap for visualizing large-scale image datasets. They employed hierarchical cluster structures to provide multi-level exploration of image datasets. Gou et al. [16] combined disentangled representation learning and semantic adversarial learning techniques to develop VATLD, a visualization system that assists in evaluating and understanding the effectiveness of Traffic Light Detection in image datasets. Similarly, Wang et al. [46] introduced DRAVA, representing image datasets as a set of small multiples and using a concept-driven approach to help users better analyze and resolve mismatches in the understanding of concepts between humans and models in disentangled representation learning. Xie et al. [49] utilized convolutional neural network (CNN) methods to generate

descriptive captions for images, enabling multi-scale exploration and analysis of large image collections. Another study [6] generated adversarial examples for each image in the dataset and proposed a visual analysis method to interpret the reasons for the misclassification of adversarial examples through multi-level visualizations. In addition, in the context of the driving domain, AutoVis [23] addressed this gap by combining a non-immersive desktop interface with a virtual reality view, enabling mixed-immersive analysis of Automotive User Interfaces.

In addition to analyzing existing image datasets, exploration of generated images has also been a focus. Feng et al. [13] jointly embedded the results of images generated by text-to-image generation models and the recommended keywords, supporting personalized exploration by users. Compared to previous works on image dataset exploration, our approach further considers the systematic generation and evaluation of image datasets through interactive exploration. This method is user-oriented and leverages original, real image datasets as a foundation to generate outcomes anticipated by the users.

3. Observational Study

In this section, we summarize the experts' conventional practices and the bottleneck of creating autonomous driving image datasets, and further distill their needs and expectations.

3.1. Experts' Conventional Practice and Bottleneck

We collaborate with four domain experts in the autonomous driving field. They include three seasoned professionals from the automotive industry (E1, E2, E3) and a researcher specializing in autonomous driving algorithm design (E4). E1 (male, age: 30) and E2 (male, age: 28) are experts in 2D/3D visual perception for autonomous driving, E3 (male, age: 29) is an expert in driving weather simulation, and E4 (male, age: 34) focuses on image generation for data augmentation through algorithm design. Each member of this group has dedicated over four years to their specialized fields.

Through discussions and research with experts, it has been confirmed that the mainstream methods for creating autonomous driving image datasets primarily involve the collection of real-world driving data. This process is primarily executed through field-based data gathering, wherein vehicles equipped with various sensors navigate through diverse driving environments to capture a wide range of driving scenarios. This raw data collection is then extensively supplemented by manual labeling processes, carried out by human annotators. These annotators identify and tag various elements within the images, such as vehicles, persons, road signs, and lane markings, to create a richly annotated dataset that is crucial for the training and validation of autonomous driving systems. This manual annotation is a labor-intensive

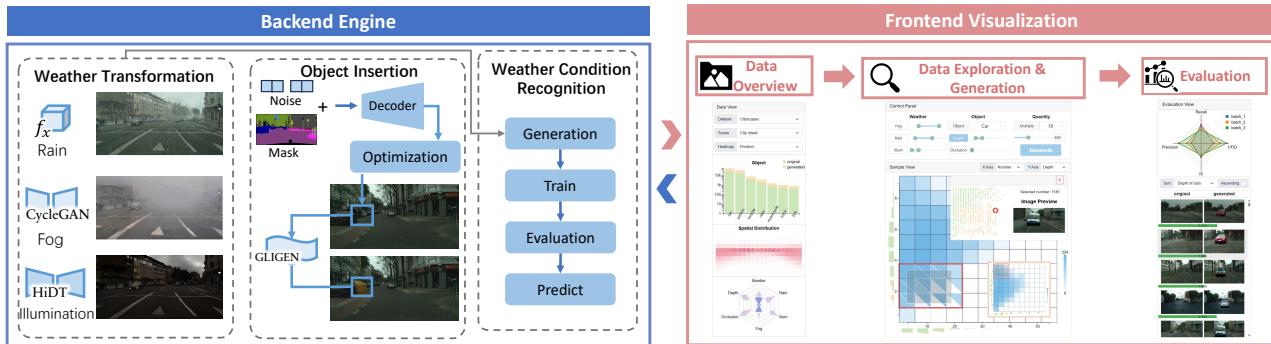


Figure 2: The pipeline of *HuGe*: in the back-end generation engine, three main modules are applied for weather transformation, object insertion, and weather condition recognition; in the front-end visualization, four coordinated views are designed for image exploration, generation, and evaluation.

and time-consuming task, requiring a high degree of precision to ensure the accuracy and reliability of the dataset. Additionally, for the acquisition of particularly uncommon instances, experts need to proactively create specific scenarios, such as deliberately placing particular obstacles on a road. Although these approaches can supplement some datasets, domain experts all agree that conventional practices often pose challenges in terms of controllability and incurring time and labor costs. Within the ambit of data acquisition, these experts encounter the following challenges. A primary concern is the intensive labor and significant time investment demanded by the manual annotation process, which stands as the most resource-consuming aspect of compiling datasets. The emergence of data generation technologies that naturally include annotations presents a promising avenue to significantly reduce these resource expenditures. Furthermore, the data collection phase, although seemingly straightforward, often requires experts to comb through extensive volumes of non-relevant data to identify the precise datasets needed. This highlights an urgent need for improved control mechanisms in the data acquisition process to streamline and target data collection more effectively. Additionally, E3 brought attention to the utilization of laboratory-based virtual simulations for generating data on uncommon scenarios where experts need to carefully design and set up controlled environments through manual intervention in real-world laboratory. While this method is effective in capturing rare events, its implementation requires high costs, further emphasizing the need for cost-efficient solutions.

3.2. Experts' Needs and Expectations

To ensure the alignment of our approach with the overarching tasks and requirements within the field, we further conducted interviews with experts (E1-E4). These interviews aimed to pinpoint their primary concerns regarding the development of controllable image generation. Through the iterative design process and our interactions with the experts, we distilled the following design requirements.

R.1 Provide an overview of the selected image dataset.

It is important to obtain a comprehensive assessment of the dataset from a holistic perspective, which can provide insights into the dataset. For example, the distribution of

what categories are in the dataset, what kinds of samples are dominant, and if lack of some important samples. Experts mentioned that this information can provide guidance for exploring possible corner cases and generating controllable corner cases.

R.2 Explore possible corner cases in the image dataset.

Corner cases are generally dangerous and novel, difficult to judge by a single indicator, and it is challenging to effectively classify corner cases from a data set. Experts seek intuitive and rapid identification of potential corner cases through visualizations that aid in analysis and evaluation. Therefore, there is a need for visualization and interaction design to help users efficiently identify potential corner cases.

R.3 Generate controllable corner cases based on the image dataset. In addition to the corner cases inherent in the dataset itself, some corner cases may arise in samples not covered by the dataset, such as those involving rainy or foggy conditions. Furthermore, existing datasets cannot encompass all possible road conditions. To address these issues, experts hope that our system can offer an interactive, human-controllable sample generation technique. This system would be capable of generating samples in bulk according to specified requirements, thereby expanding the sparse areas of the dataset.

R.4 Evaluate and compare generated image samples.

The samples generated may exhibit certain issues, necessitating a further assessment of their quality with the expertise of professionals. Moreover, experts are required to compare the outcomes of images generated through various human-controlled parameters to ascertain which parameters yield superior results. The system needs to incorporate comparative views, enabling experts to inspect and evaluate the generated samples, thereby ensuring the quality and applicability of these samples.

4. Approach Overview

We propose *HuGe*, an interactive visual analytics system for generating controllable images based on the proposed generation method that combines traditional optimization algorithms and generative models. Fig. 2 shows the pipeline of *HuGe*, which consists of a *back-end generation engine* and



Figure 3: The example effect of changing the weather conditions. The horizontal axis represents different types of weather control, namely illumination, fog, and rain, while the vertical axis indicates their increasing intensity.

a *front-end visualization*. In the backend generation engine, three integral modules are implemented: weather transformation, object insertion, and weather condition recognition. The weather transformation module facilitates meticulous manipulation of weather parameters such as rainfall intensity, fog density, and illumination, affording users precise control over environmental elements. The object insertion module empowers users to seamlessly integrate diverse objects into a curated selection of images, enhancing the versatility and customization of generated content. Furthermore, weather condition recognition is designed to identify and quantify the severity of weather conditions, enabling users to select images from the original dataset that match their interests in terms of weather intensity. In the front-end visualization, four coordinated views are designed for image exploration, generation, and evaluation. To be specific, the data view (Fig. 1A) gives an overview of the selected dataset. This high-level summary allows users to quickly assess the composition and characteristics of the dataset before proceeding to more granular exploration (**R.1**). The control panel (Fig. 1B) allows users to conduct fine-grained, parametric control over a multitude of factors, including global environmental conditions and granular object-level attributes (**R.3**). The sample view (Fig. 1C) enables users to explore the distribution of the selected data in greater granularity across two user-specified attributes, which provides in-depth exploration of possible corner cases (**R.2**). The evaluation view (Fig. 1D) performs a comprehensive evaluation of the synthetic outputs, analyzing at both the aggregate dataset level as well as the granular instance level, which shows evaluation metrics across four widely adopted indexes and visual comparisons between the original seed images and their corresponding synthetic counterparts (**R.4**).

5. Back-end Engine

In this section, we describe the methods and models adopted for weather transformation, knowledge-based

controllable object insertion, and intensity recognition for weather conditions in the back-end engine.

5.1. Weather Transformation

When modifying image environments, domain experts often prioritize illumination, fog, and rain as key factors due to their frequent occurrence and significant impact on driving conditions [36]. Therefore, our focus in this study is on transformations related to these factors (**R.3**), without considering conditions such as snow and hail, as they represent a practical starting point for addressing environmental variability in autonomous driving scenarios. The example effect of changing the weather conditions is shown in Fig. 3.

For illumination transformation, we adopt the high-resolution daytime translation model (HiDT) [3] which can produce high-resolution images with variations in illumination. Using HiDT, we achieve realistic illumination modifications across various images.

For fog transformation, we adopt CycleGAN [54], a prototypical Conditional Generative Adversarial Network (CGAN), which can provide precise control over the generative process. It requires solely input images and associated degrees of fog, to generate the synthesis of authentic foggy imagery.

For rain transformation, unlike illumination and fog, there is a scarcity of annotated datasets specifying rain intensity, complicating the use of generative models like GAN for controlled rain generation. MagicDrive [14] effectively generates realistic rain effects, yet its approach, while rendering accurate rain patterns, often alters the quantity, position, and color of objects in the image. Additionally, control over rain intensity is achieved only qualitatively through verbal cues, lacking the desired quantitative precision. In our study, to maintain object characteristics in driving scenes and achieve precise control over rain intensity, we adopted the approach proposed in [45] for rain generation. This method can produce varying degrees of realistic rain images.

5.2. Knowledge-based Controllable Object Insertion

As shown in Fig. 4, we utilize optimization algorithms and diffusion models to controllably add objects to images based on users' expectations. Specifically, we employ a conditional variational autoencoder (CVAE) [42] to derive candidate boxes. These boxes are then optimized through an iterative process that incorporates predefined rules—such as aspect ratio, size limits, and alignment with the road surface—and user-specified adjustments, such as modifying the position, depth, or occlusion percentage. This iterative optimization refines the bounding boxes to meet both the constraints and user-defined preferences, providing fine-grained control over various aspects of the generated boxes. Afterward, we employ a diffusion model, the GLIGEN [30] model, to introduce user-specified objects into the original image.

Short Title of the Article

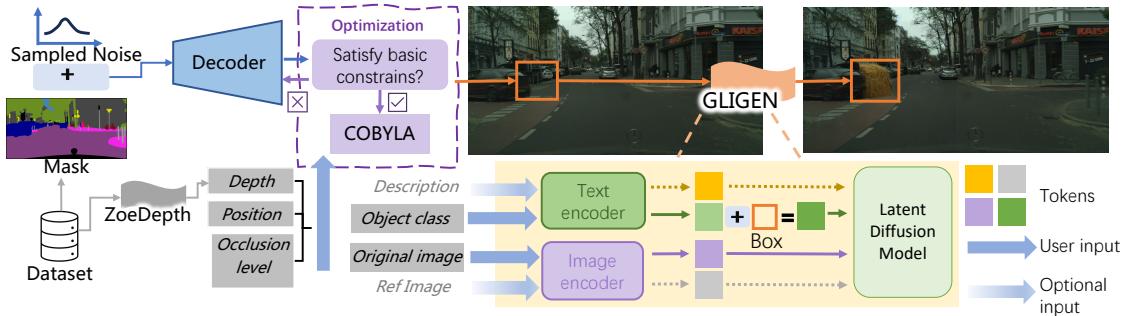


Figure 4: The workflow of the knowledge-based controllable object insertion: the CVAE model is first used to derive candidate boxes, optimizing them based on predefined rules and user adjustments; given the candidate boxes, the GLIGEN model is then utilized to introduce user-specified objects into the original image.

Initially, similar to the approach proposed in [18], we also use the CVAE model to extract latent features and generate candidate obstacle positions through context-aware spatial representation learning. The resultant bounding boxes are intricately linked to the semantic information of the images, ensuring a more coherent fit with image semantics compared to the generation of arbitrary positions. However, we find that relying solely on semantic segmentation maps without depth information can lead to unreasonable occlusion order when adding obstacles based on the sampled latent vectors, e.g., an object that should be occluded appears in front of the occluding object. In addition, the box sizes generated by this method may not match the actual object sizes, and it cannot generate novel unseen objects. Furthermore, the physical meanings represented by certain latent vector dimensions are rather difficult to comprehend, preventing the desired controllable generation.

To address the issues of occlusion order and object size, we incorporate depth information into the context-aware spatial representation learning framework. As shown in Fig. 4, this methodology utilizes a depth estimation model, specifically ZoeDepth model [5], which has been fine-tuned on the KITTI [33]—an authoritative dataset in autonomous driving research—to extract and integrate relative depth data. Leveraging foundational principles of imaging and projective geometry, assuming no lens distortion and the actual height and width of an object are (h_o, w_o) , its distance from the camera is d , then its height and width (h_i, w_i) in the image relate to d by $h_i = \frac{f \times h_o}{d}; w_i = \frac{f \times w_o}{d}$. Here f is the focal length. Hence the area S_i of the object box in the image and its physical area S_o satisfy $S_i = \frac{f^2}{d^2} S_o$. It follows that the object dimensions in the image are inversely proportional to the square of depth. For objects in the original dataset, we can estimate their real size distribution by aggregating statistics of the product of their width, height, and depth. Therefore, given the average depth of the box edges, we can compute the distribution of (h_i, w_i) and filter out unreasonable candidates accordingly. For new objects outside the dataset, users can define their aspect ratio relative to a reference object to obtain the size distribution. This allows for ensuring plausible occlusion orders and object sizes. Similarly, for occlusions, we compare the depth of

the candidate bounding box with that of intersecting objects, constraining that the depth of the occluding object's bottom edge is not less than that of the occluded object's bottom edge. This ensures that the bounding box is positioned correctly in space.

Furthermore, acknowledging the inherent characteristics of obstacles and common scenarios, we impose constraints on the aspect ratio of obstacles and constraints related to the road surface. Users have the flexibility to add constraints to the candidate bounding box based on their specific requirements, such as constraints on the percentage of occlusion, position, or depth. For instance, in the scene shown in Fig. 4, the straw was assigned a greater depth value (i.e., placed farther away) than the car based on the estimated depth map. The optimized bounding box reflects this specific depth requirement while preserving the object's natural relationship with the road surface. Considering the non-linear and constraint-heavy nature of the optimization problem, we adopt the COBYLA optimization algorithm [38] due to its ability to handle non-linear constraints effectively and its balance between performance and runtime efficiency. To streamline the optimization process, we define a set of simple and easily satisfiable constraints, including aspect ratio and size, as the foundational constraints. It is mandated that the generated bounding box must adhere to these base constraints before initiating the optimization algorithm for resolution. This strategic approach serves to mitigate the complexity of optimization and enhances the feasibility of achieving satisfactory results.

Finally, we adopt the GLIGEN [30] model as the generation backbone, due to its strong zero-shot capability and its flexible design for incorporating multiple conditioning inputs. In our study, we specifically utilize GLIGEN's inpainting mode, which not only preserves the surrounding context but also allows the model to naturally reason about spatial coherence, lighting, and texture consistency. As illustrated in Fig. 4, the model synthesizes the final completed image based on the given bounding box, object category, and original image, along with optional inputs such as textual descriptions and reference images.

5.3. Intensity Recognition for Weather Conditions

Recognizing the intensity of different weather conditions in selected images could provide information for exploring and comparing the original image dataset and the generated image dataset. Given that prevailing autonomous driving datasets typically annotate weather types but lack detailed information on the intensity, we conduct training on the recognition of rain, fog, and illumination. To ensure consistency between the model's results in intensity recognition and the generated intensity scale, we apply the controllable weather generation models mentioned in Section 5.1 to transform images in the dataset. Using the controllable weather generation models' output conditions as labels, we successfully constructed a dataset containing 59,500 images with information on the intensity of rain, fog, and illumination based on the trainset of Cityscapes. Building upon this dataset, we train three ResNet [17] models specifically dedicated to the task of rain, fog, and illumination intensity classification. Then, we evaluate the performance of the three ResNet models, and their mean squared errors on the test set are consistently within 0.05, which can provide satisfactory results for the following tasks.

6. Front-end Visualization

We design and implement a visual analytics system to explore, generate, and evaluate image samples, as shown in Fig. 1. The system consists of four views: the data view, the control panel, the sample view, and the evaluation view.

6.1. Data View

The data view (Fig. 1A) contains four components, including widgets for configuring datasets (Fig. 1A₁), a stacked bar chart showing the number of objects (Fig. 1A₂), a heatmap showing the spatial distribution of objects (Fig. 1A₃), and a radar chart with violin plots showing the distribution of different dimensions (Fig. 1A₄), which provides an overview of the selected dataset. After users select a dataset, the corresponding information will be updated. Users can select specific scene within the dataset to conduct in-depth exploratory analysis on the selected scenario. To provide information about what objects are in the image dataset, the stacked bar chart (Fig. 1A₂) is used to display the count of different objects. In this chart, green color represents the original image data, while the orange color represents the generated image data. Given the significant differences in the quantity of various objects (for instance, the number of trucks is much less than that of cars), a logarithmic function is used to adjust the scale of the y-axis. Furthermore, a heatmap is used to display the depth and position of different objects in the selected dataset (Fig. 1A₃). The x and y axes of the heatmap correspond to the width and height of the image samples, respectively, where redder areas indicate higher object overlap rates in those regions. Lastly, a radar chart with violin plots (Fig. 1A₄) is integrated to perform statistical analysis across multiple dimensions of the overall image dataset, including the number of objects, the depth of objects, the degree of occlusion, and weather conditions in

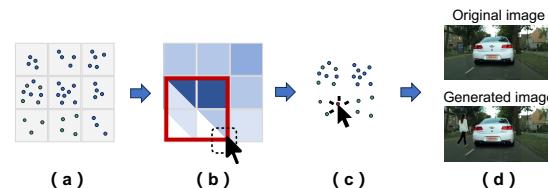


Figure 5: Current design of the sample view. (a) Divide samples into different grids and count the number of samples in each grid to draw the heatmap matrix. (b) Select samples in a region of interest. (c) Hover to check selected samples. (d) The image of the corresponding samples will be displayed in Fig. 1C₅.

the images, like rain, illumination, and fog. The sector part in the middle shows the KL divergence value between the distribution of each dimension and uniform distribution. After calculating KL values, then a normalization is performed. Large areas indicate a larger uneven distribution.

6.2. Control Panel

To provide human-controllable and quantifiable settings in the generation process, we designed the control panel (Fig. 1B), which contains two main functional modules: one for adjusting weather conditions (Fig. 1B₁) and the other for adding new objects (Fig. 1B₂).

To start the generation process, first, users are required to select a group of samples and decide how many new samples need to be generated in the sample view (Fig. 1C). For adjusting the environment, users can adjust the fog, rain, and illumination ("Illum") levels of the dataset in Fig. 1B₁, ranging from 0 to 1. For adding new objects, users can select different objects, including cars, trucks, persons, and so on. Users can then adjust the distance ("Depth") and occlusion of the objects ("Occlusion") in Fig. 1B₂. To allow new objects to appear in a certain area, users can select the area of interest in Fig. 1A₃. Finally, users can click the "Generate" button to generate a new batch of samples.

6.3. Sample View

The sample view (Fig. 1C) is the primary view for users to explore the image collection. It aims to assist users in exploring the distribution of the data set along certain dimensions with more details. Users can select two dimensions as the x-axis and the y-axis correspondingly in Fig. 1C₁. The heatmap matrix in the middle (Fig. 1C₂) demonstrates the distribution of the dataset in two selected dimensions and each grid of the heatmap matrix contains image samples. The histograms along the two axes (Fig. 1C₃) display the distribution of the dataset in one dimension. Users are allowed to select data samples in the heatmap matrix to further explore the distribution of samples. Considering that the user's primary intention is to fill in sparse or missing areas using the original dataset, our design focuses on showing the distribution of both the existing and generated images rather than just the generated images. When users complete the selection, the data samples will be highlighted, and a pop-up window will display a sample scatter plot (Fig. 1C₄). The x-axis and y-axis of the scatter plot are consistent with two

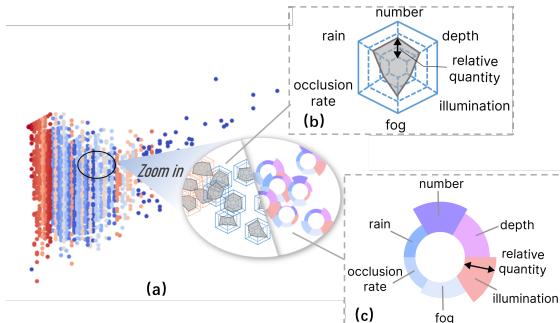


Figure 6: Alternative designs for the sample view. (a) Display a PCA-reduced scatter plot with DBSCAN clusters, using color to differentiate clusters. Zooming in reveals details via (b) a radar chart, detailing a point's metrics across six dimensions, with color indicating cluster membership; or (c) a glyph, using six sectors to represent dimensional information, with the sector's height representing relative quantity.

dimensions selected by users, with the range of the two axes matching the selected area. The green dots mean samples from the original dataset, while the yellow dots indicate generated samples. As shown in Fig. 5c, users can hover over dots and the image of the samples will be displayed in Fig. 1C₅. After generating the image samples, each corresponding grid in the heatmap matrix will be divided into two triangles (Fig. 1C₆). The left-bottom part represents the original image data in that area, while the right-top part shows the distribution of original image data combined with the batch-generated images.

Alternative design. During the iterative development process, we explored two alternative designs for the sample view. Initially, we intended to use scatter plots to display information across all six dimensions. To achieve this, we first applied PCA to reduce the dimensionality of the data to a two-dimensional plane, followed by clustering using the DBSCAN algorithm. In the scatter plots, the x and y axes represent the two dimensions obtained through PCA, with point colors indicating cluster categories, as shown in Fig. 6a. To present detailed information on the six dimensions, we designed an approach where points would transform into radar charts or glyphs upon zooming in, as illustrated in Fig. 6b&c. In the radar chart (Fig. 6b), the six axes correspond to the relative information on the six dimensions, with color representing the cluster category. In the glyph representation (Fig. 6c), six sectors each represent a dimension, where the sector's height indicates the relative information level. However, due to large number of images, there was significant overlap between glyphs and radar charts, making data unclear. Additionally, the PCA-driven approach reduced dimensionality but obscured critical details needed for setting specific parameters. Ultimately, we did not adopt this design option.

6.4. Evaluation View

The evaluation view (Fig. 1D) is designed for analyzing and comparing the generation results, which consists of

two parts, a radar chart showing four evaluation metrics (Fig. 1D₁) and an image list comparing generation results (Fig. 1D₂).

Firstly, for the batch of image samples, we have selected four evaluation metrics: Fréchet Inception Distance (“FID”), Inception Score (“IS”), Precision, and Recall. FID focuses on the overall similarity between generated and real images, while IS evaluates the quality and diversity of individual images. A lower FID score indicates higher image quality and realism, whereas a higher IS score suggests good quality and diversity. Both metrics are often used together to assess the performance of generative models. Additionally, we adopt the improved precision and recall metric [26], where precision gauges how closely generated images resemble real data, and recall measures the diversity of generated images in capturing the real data distribution. Specifically, in the context of generative models, the interpretation of precision and recall changes. Precision measures the proportion of generated images that are “realistic”, that is, those that closely mimic the real data distribution. A high precision implies that the majority of images generated by the model are indistinguishable from those in the actual dataset. As for recall, it assesses the diversity of the generated images, or in other words, the extent to which the model captures the real data distribution. A high recall indicates that the model can generate a broad range of images that convincingly resemble those from the real dataset. In summary, an ideal image generation model should possess low FID, high IS, high Precision, and high Recall scores, indicating the model’s capability to generate images that are both realistic and diverse in high quality. We present these four metrics in the form of radar chart and normalize them to the same scale in Fig. 1D₁. It is important to note that only the FID metric is better when lower, while metrics like IS are better when higher. Therefore, for FID, we use the reciprocal form, i.e., 1/FID. To showcase real images and compare them before and after generation, we present the batch of image samples as an image list. Additionally, we visualize the selectable sort values from the dropdown menu as green bars. For example, in Fig. 1D₂, the length of these bars indicates the depth of the cars. Users can select different dimensions for sorting and comparing the effects of images before and after generation side by side, which can provide intuitive guidance for the subsequent image generation.

7. Evaluation

We conducted two case studies, a metric-based evaluation and interviews with E1-E6 to demonstrate the effectiveness and usability of our system. The background of E1-E4 has been introduced in Section 3.1. E5 (female, age 30) and E6 (male, age 29) are newly invited experts who have a strong expertise in visual perception for autonomous driving. Section 7.1 and Section 7.2 present the cases conducted by E1 and E3 respectively. Section 7.3 presents a metric-based evaluation of our method’s impact on improving the

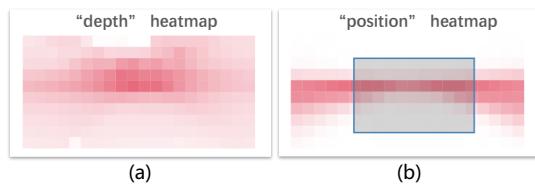


Figure 7: (a) shows car distribution in the “depth” heatmap, while (b) shifts focus to the “position” heatmap for car placement. The manually selected range is highlighted in blue.

detection model’s performance. Section 7.4 presents the feedback from all experts.

7.1. Case I: Controllable insertion of cars and persons

E1 focuses on improving object detection in autonomous driving and often needs to test models on different scenarios, particularly those involving potential hazards. In this case, he would like to explore the original dataset and generate some images that may present higher risks for testing. Moreover, he believed that acquiring images of target objects at varying depths and positions in real-life conditions presents significant challenges. Consequently, he aspires to freely manipulate images in these aspects for more comprehensive analysis. At first, E1 selected the Cityscapes dataset [9], a widely used object detection dataset, for preliminary exploration. This dataset, consisting of densely pixel-annotated urban landscapes across 19 categories, captures street scenes in 50 cities. As shown in the bar chart (Fig. 1A₂), he noticed “car” and “person” were the two most abundant categories. These also are crucial recognition targets in autonomous driving scenarios, hence they were selected as subjects of analysis. When clicking on the bar corresponding to “car”, he found the color of center area in the “depth” heatmap was darker than other areas (Fig. 7a), which indicated “car” in the middle was far away. Also, switching to the “position” heatmap, he found the color of center area was lighter than other areas (Fig. 7b), which indicated most cars were positioned along the roadsides and few cars ahead at close range. In densely populated urban areas, close proximity of cars is common and poses a high risk. Therefore, E1 selected the area using a blue rectangle (Fig. 7b) and decided to add cars in this area to simulate high-density urban traffic and address dataset shortcomings. Then, to select base images for desired image generation, he turned to the radar chart (Fig. 1A₄) for analyzing different dimensions. To insert cars, E1 placed greater emphasis on dimensions more pertinent to the insertion of cars and persons, rather than on weather-related aspects. He noticed significant KL divergence in the number dimension but less so in depth. Hence, he selected “number” as the x-axis and “depth” as the y-axis, the distribution of the original image dataset was shown in the sample view (Fig. 1C). Considering image generation with less occlusion, he decided to select images with less number and large depth of cars as the base images, as shown in the position of the red box in Fig. 1C₇. Further, in the control

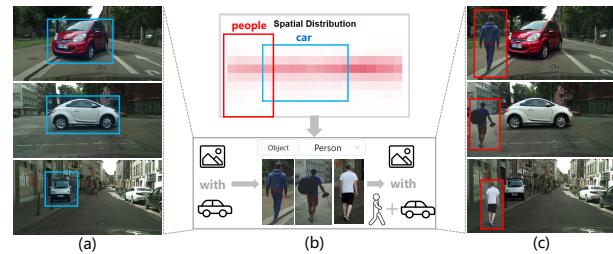


Figure 8: Adding “person” additionally to images that already have cars inserted. (a) the generated image after adding “car” based on the original sample image from Cityscapes. (b) the position distribution of persons, where we first select the area on the left and then proceed to add “person”. (c) the generated image after adding “person” again.

panel (Fig. 1B), he entered the prompt word “car” and gradually adjusted the depth, choosing distances of [1,3], [2,4], [3,5] as parameters for three rounds of generation. The generation multiplier was set to the default 1X. After pressing the generate button, desired images are generated. The semi-automatic bounding box generation process exhibited an average computational time of about 0.8 seconds per image, demonstrating an acceptable performance in augmenting the dataset. Taking the distance range of [1, 3] as an example, at the position of the sample (Fig. 1C₆), the original rectangle has transformed into two triangles, significantly filling the previously white (data-sparse) area. Additionally, as shown in Fig. 1C₃, the newly generated image data appeared in the form of small yellow rectangles. Furthermore, E1 compared the original and generated images by clicking on the newly generated yellow dots in Fig. 1C₄ and the corresponding changes in Fig. 1D₂. Upon a meticulous examination of all three batches of the generated images, E1 confirmed that all synthesized objects adhered to the specified requirements and exhibited contextually an appropriate spatial layout, ensuring the integrity and relevance of the augmented dataset. After that, E1 moved to the evaluation view (Fig. 1D) to compare the three batch generated images. The scale on the right axis represented the value of 1/FID. When the right endpoint of the quadrilateral moves further to the right, the value of FID is lower, indicating a closer statistical resemblance to real images. Firstly, based on the distribution of the right vertices of quadrilaterals in Fig. 1D₁, the results of the third batch were closer to the origin, indicating that the FID value of the third batch was higher than those of the previous two generations. Secondly, the scale on the left axis represented the value of precision. The higher precision value suggested that the generated images were closer in quality to real images. The expert believed the generated objects might have less impact on the overall image because they are farther away, appear smaller, and occupy fewer pixels. Thirdly, the IS and Recall values of the three batches were similar, indicating no significant change in image diversity. After the comparison, E1 thought the third batch had better quality. Taking everything into account, E1 decided to incorporate the third batch of data into the original dataset.

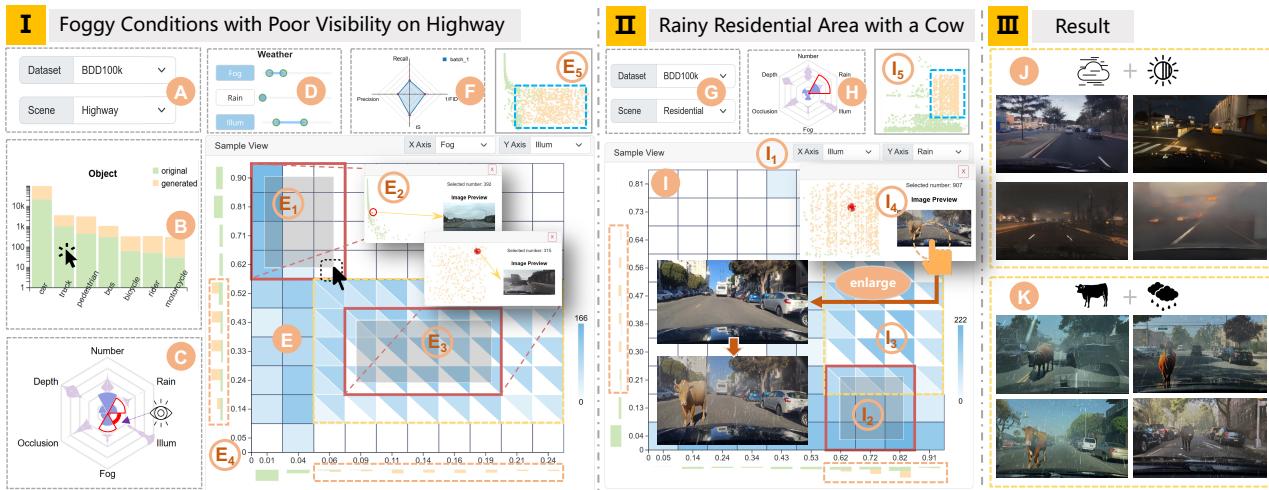


Figure 9: Experts workflow for controllable generation of corner cases on the BDD100K dataset. (I) Foggy Conditions with Poor Visibility on Highway. (II) Rainy Residential Area with a Cow. (III) showcases examples of the final generated images; the upper part is the result of process (I), and the lower part is the result of process (II).

Similarly, E1 would like to generate images with “person”. He then selected “person” as the subject for further analysis in the statistical bar chart (Fig. 1A₂). Observing the spatial distribution heatmap (Fig. 8b), he noted that there was a higher concentration of persons on the right side compared to the left. Consequently, he aimed to add more persons to the left side of the images. Additionally, he considered that to generate dangerous scenario, he selected the first batch of images with cars relatively close to the original image (Fig. 1C₆). By adjusting the slider (Fig. 1B₂), he chose 100 images to add “person”. In the end, he obtained the desired result, as shown in the example in Fig. 8c.

In summary, E1 found that the operation workflow of *HuGe* is intuitive and straightforward, enabling the creation of image data that is usually difficult to capture through conventional road testing with cameras. He stated: “*Although some objects in the images might not be completely realistic, they nonetheless contribute to enriching the dataset for object detection model testing. Coupled with visual analysis, this represents a meaningful endeavor.*”

7.2. Case II: Controllable generation of corner cases

In this case, we collaborated with E3 to systematically explore the capability of *HuGe* in generating controlled corner cases for autonomous driving. His primary focus was on the effectiveness of *HuGe* in producing images of extreme situations, particularly those involving low visibility due to severe weather conditions and the presence of uncommon objects on the road. Drawing from 5 years of weather simulation experience, E3 identified significant limitations in existing autonomous driving datasets regarding extreme weather conditions, citing their rarity, unpredictability, and high collection costs. Consequently, E3 opted to utilize *HuGe* for generating and adjusting weather conditions in these corner case scenarios. At first, E3 selected the BDD100K dataset [51] for exploring weather conditions in image

datasets, as this dataset includes scenes under various times and weather conditions. Then, he noticed that under the highway category, the number of trucks ranked second, as shown in the Fig. 9B. “*Trucks are primarily used for cargo transportation on highways.*” he said. He mainly focused on the weather and considered fog, rain, and illumination. He noticed that in the radar chart (Fig. 9C), the tail of the illumination dimension was quite wide, indicating that most images were in relatively bright environments. Thus, he chose illumination as the Y-axis. Then, since the KL divergence of fog was slightly higher than that of rain, he decided to choose fog as the X-axis for exploration. Besides, he wanted to generate more images with a small range of fog to add more possibilities, making the fog dimension smoother and more uniformly distributed. Integrating the information observed earlier, he explored and analyzed the dimensions of fog and illumination in the sample view (Fig. 9E). The heatmap distribution and the height of the bars on the axes confirmed his analysis (Fig. 9E₄). He then selected the upper-left section (Fig. 9E₁), which represented highways with high illumination and low fog, meaning higher visibility (Fig. 9E₂). E3 wanted to transform and generate images with lower visibility (decreasing illumination and increasing fog), targeting the originally vacant bottom-right area in the sample view (Fig. 9E₃). Therefore, in the control panel (Fig. 9D), he adjusted the fog to [0.05, 0.35] and the illumination to [0.1, 0.5]. Upon completion, the evaluation metrics results were shown in (Fig. 9F). He observed the single batch and noticed that the values of all four metrics were not high, which he considered that the domain variation issues introduced by the models generating fog and illumination were responsible, but the overall generated effect was acceptable. Besides, he observed the part outlined in yellow dashed lines in Fig. 9E₄, which precisely corresponded to the region he intended to supplement. Fig. 9E₅ represents the distribution of image samples across the entire visible area of the sample view, with the generation range desired

by E3 outlined in a blue dashed line. Eventually, the style of the images he generated was shown in Fig. 9J.

After using *HuGe* to adjust weather conditions and acknowledging the system's capability to insert any object, E3 proposed the idea of controllably generating foreign objects in challenging weather conditions. According to E3, this idea was sparked by his own experience of encountering a cow in a residential area while driving, an event he deemed highly improbable yet verifiably real. This type of scenario, involving unusual behaviors of other road users (e.g., animals crossing the road), is precisely the kind of corner case we highlighted as crucial in the Introduction section. This spurred his interest in using *HuGe* to generate similar and uncommon images. First, he selected "Residential" as the scene (Fig. 9G). Subsequently, in the radar chart (Fig. 9H), he observed that "rain" exhibited the highest KL divergence, indicating the greatest imbalance. Thus, in the sample view, he selected "Illum" as the x-axis and "Rain" as the y-axis (Fig. 9I₁). He selected a batch of images depicting clear weather from the bottom right corner as the base (Fig. 9I₂). He entered "cow" in the object bar to specify the object type. Based on his understanding of the random appearance of foreign objects, he adjusted the degree of occlusion ("Occlusion") in the control panel to not exceed seventy percent and chose a distance range of [2,4], without specifying a location. Moreover, aiming to fill the relatively vacant area by moving the original region upwards (Fig. 9I₃), as indicated by the rectangular distribution in the sample view, he adjusted "Rain" to [0.2, 0.6]. Next, the yellow dots in Fig. 9I₄ illustrated the distribution of the generated images. By hovering over the dot with the mouse and clicking to select the image, he observed the enlarged original and generated images on the left side. Fig. 9I₅ and Fig. 9E₅ convey similar meanings, with the blue dashed line indicating the area where triangles appear in the sample view of Fig. 9I. The generation samples are shown in Fig. 9K.

In this case, E3 was notably impressed with the capabilities of *HuGe* for generating conditions of extreme weather or rare and realistic scenes. E3 successfully identified and filled the data gaps in extreme weather conditions by analyzing and manipulating the BDD100K dataset. He noted: "*The coordinated design of the sample view helps me quickly identify which dimensional areas are missing or sparse, and after generation, I can see the filled areas.*" E3 also enhanced the diversity and practicality of the dataset by generating uncommon images, such as scenes of encountering cows while driving in the rain. He thinks that *HuGe* offers a comfortable and logically clear interaction experience, with the implemented functionalities being quite exciting.

7.3. Metric-Based Evaluation

To evaluate the efficacy of our generated images in advancing model performances for specific target scenarios, we conducted a metric-based evaluation of our method's impact on improving the detection model's performance. This evaluation addresses the close-range sample deficiency

in the Cityscapes dataset identified in Section 7.1. By augmenting the dataset with our generated images, we aimed to improve the overall training performances, as measured by the metric mean Average Precision (mAP).

Training set. The Cityscapes dataset is a widely recognized benchmark for semantic segmentation in autonomous driving scenarios, comprising 19 semantic classes. For our object detection task, we selected 4 categories of different sizes of traffic-relevant objects, including cars, persons, bicycles, and trucks—encompassing three different vehicle sizes and a category of human. We transformed the segmentation annotations into bounding boxes, effectively converting them into an object detection dataset suitable for our purposes. Therefore, we utilized 2,975 training images from the Cityscapes dataset as a base training set.

Fintuning set. We meticulously prepared three distinct finetuning datasets to comprehensively evaluate the usability of *HuGe* generated data: CARLA Dataset, *HuGe* Dataset and the selected Cityscapes Dataset. For the CARLA dataset, we selected 827 images from the CARLA Detection dataset[10], focusing on close-range vehicular environments. These images were strategically extracted to represent the most proximal range, ensuring relevance to near-distance object detection scenarios. The CARLA Detection dataset itself was generated using the CARLA simulator, with data collected in autopilot mode across diverse environments, including Town01, Town02, Town03, Town04, and Town05. For the *HuGe* dataset, We generated 827 synthetic images through the initial process of Section 7.1, specifically simulating close-range driving scenarios. These meticulously crafted images were derived from our training set and served as supplementary fine-tuning data, strategically designed to enhance the model's object detection capabilities in near-proximity contexts. Lastly, to address potential bias in finetuning result produced by the original Cityscapes in creating *HuGe* dataset, we curated the selected Cityscapes Dataset through a refined selection process. This involved creating a specialized 827-image subset from the Cityscapes datasets, deliberately selecting the foundational images used in the initial generation process.

Validation set. We prepared two validation sets for evaluation: the Cityscapes validation set and the specialized close-range validation set. The standard Cityscapes validation set of 500 images demonstrates model performance in common scenarios, providing a baseline for comparison. The close-range validation set, also containing 500 images, focuses on close-range scenarios to accurately assess model performance in scenarios similar to those identified in Section 7.1. Here, we constructed the specialized close-range validation set, which selected 500 images from the BDD100K validation set that aligned with the conditions of being close and in the center. This approach allowed us to more precisely evaluate model performance in the specific close-range scenarios of interest.

Procedures. Our experimental setup involved training two classic and widely used object detection models: YOLOv5 [24] and Faster R-CNN [40]. First, we trained

each model for 300,000 iterations on the Cityscapes training set to serve as base models. Then, we employed a transfer learning approach, fine-tuning the models (pre-trained on the Cityscapes dataset) using each fine-tuning set for 3,000 iterations, including the CARLA Dataset, the *HuGe* Dataset, and the selected Cityscapes Dataset. After fine-tuning, we validated the base models of Faster R-CNN and YOLOv5 on the Cityscapes validation set and the close-range validation set. Additionally, we evaluated the fine-tuned models (finetuned on the three fine-tuning sets) on the close-range validation set.

Results. In Table 1, we compared the performance of Faster R-CNN and YOLOv5 on the original Cityscapes Validation set and the Close-range Validation set. Both Faster R-CNN and YOLO is trained only on the Cityscapes training set. As illustrated in Table 1, the base models' detection accuracy on the selected close-range validation set was substantially lower compared to the original Cityscapes validation set. This insight confirmed the performance gap in the models trained on the Cityscapes dataset, particularly when detecting objects in close-range scenarios.

As further detailed in Table 2, we conducted a comparative evaluation of the model performance on close-range validation set when finetuned with the CARLA dataset, the selected Cityscapes datasets and the *HuGe* dataset on the Close-range Validation set. For YOLOv5, the highest mAP values were achieved when fine-tuned on the *HuGe* dataset, with significant gains observed across all object categories. Specifically, the mean Average Precision (mAP) for car detection improved substantially across both architectures tested. For YOLOv5, the mAP increased from **34.8%** to **53.2%**. Similarly, Faster R-CNN exhibited an improvement from **38.4%** to **42.8%**. Furthermore, both models showed considerable improvements in detecting other object classes, particularly persons and bicycles.

In contrast, fine-tuning on the CARLA dataset resulted in only modest improvements for YOLOv5 and marginal changes for Faster R-CNN, indicating limited generalizability of the CARLA data to real-world scenarios. The selected Cityscapes dataset, while offering improvements in certain categories, showed lower overall performance compared to the *HuGe* dataset. These results underscore the effectiveness of the *HuGe* dataset in bridging the performance gap for the detection of objects at close range.

Furthermore, an interesting anomaly was observed in the case of truck detection using Faster R-CNN, where performance degraded after finetuned on *HuGe*, and the best performance was achieved when finetuned on the selected Cityscapes dataset. We hypothesize that this discrepancy may be attributed to the limited number of truck instances in the dataset and the structural differences between YOLOv5 and Faster R-CNN, leading to increased variability in the performance of Faster R-CNN on this class. YOLOv5's grid-based, one-stage detection is more resilient to sparse data, while Faster R-CNN's two-stage mechanism may struggle with imbalanced classes, making its truck detection performance unstable.

Table 1

Performance Comparision of Base Models on Different Validation Set

Model	Validation Set	car	person	bicycle	truck
YOLOv5(base)	Cityscapes	70.5	46.7	34.7	33.9
	Close-range	34.8	18.1	11.3	8.2
Faster R-CNN (base)	Cityscapes	62.4	43.3	42.6	50.0
	Close-range	38.4	16.2	32.1	15.4

Table 2

Performance Comparision on Close-range Validation set of Models Finetuned on Different Fintuning Dataset

Model	Finetune Dataset	car	person	bicycle	truck
YOLOv5	None(base)	34.8	18.1	11.3	8.2
	Cityscapes	27.8	20.4	38.1	9.48
	CARLA	31.5	18.5	12.1	7.62
	<i>HuGe</i>	53.2	37.5	40.5	23.4
Faster R-CNN	None(base)	38.4	16.2	32.1	15.4
	Cityscapes	31.3	11.7	26.8	34.1
	CARLA	27.3	10.0	12.9	19.4
	<i>HuGe</i>	42.8	27.1	40.0	5.9

7.4. Expert Interview

To better evaluate the effectiveness of *HuGe*, we conducted individual interviews with experts E1-E6 (their background has been introduced at the beginning of Section 3.1 and Section 7). First, we briefly introduced our system and provided a simple tutorial demonstrating the visual design and interaction of *HuGe*. Next, experts could explore the system for about an hour. Then we conducted a half-hour individual interview with each expert and gathered their valuable feedback.

System Design and Usability. The feedback on our visual analytics system indicated that the system is “*easy to understand*” and “*simple to operate*”. For example, E6 mentioned that “*The histogram and heatmaps allow me to quickly grasp the object distribution and identify any gaps in the dataset at a glance.*” Experts found that the sample view effectively showcased the points they wanted to analyze, revealing uneven distributions of image datasets across various dimensions. E2 valued the ability of *HuGe* to allow users to select any two dimensions for analysis, observe the distribution of individual image samples within the entire dataset, and support effective comparisons. He believed this approach allows for flexibility and focused analysis, a point also mentioned by E1 and E5. Most experts appreciated the interactive functionality of heatmap brushing selection and dimension analysis. Additionally, E3 and E6 both agreed that this system can assist data analysts in exploring the dimensional features of autonomous driving images and uncovering potential issues. For instance, E3 discovered incorrect ground truth annotations for buses in parking lot scenes using the object heatmap.

Effectiveness. Experts were generally satisfied with the overall process and believed that *HuGe* can significantly reduce the time and manpower cost for high-quality image

collection or generation in the field of autonomous driving. Also, the generated images can effectively supplement the existing dataset to enhance model training. Regarding generating different weather conditions, E3 mentioned that this generation method does not require complex historical data and environmental parameters for adjustment, only the intensity of the corresponding weather needs to be adjusted. Moreover, E5 pointed out, “*Although the generation speed is not very fast, it can generate images that are difficult to capture in real-world scenarios, which can greatly save the costs of collecting image data for autonomous driving.*” Overall, experts provided positive feedback on the system’s effectiveness, believing that it can enrich the autonomous driving image dataset, especially for samples that are difficult to collect in real world.

Improvement. Experts also offered several suggestions for improving *HuGe*. Firstly, they hoped the efficiency of image generation could be enhanced to improve the timeliness of generating larger-scale image data. Secondly, improving the image generation effect with more advanced models. Thirdly, editing only a part of an image, rather than the whole, with current technologies (such as GAN models and diffusion models) can easily result in domain inconsistencies within the image, which should be iteratively considered in future work.

8. Discussion and Future Work

This section discusses several issues of *HuGe* and possible solutions for future work.

Evaluating reality in generated images. Assessing the reality in generated images involves multiple subjective considerations, such as visual fidelity, contextual coherence, and user satisfaction, which lack quantitative metrics. Attaining authentic reality requires evaluating the model’s capabilities, the diversity and quality of datasets, and grasping the nuances of human perception. While FID and IS provide some measures of image quality, they don’t capture all dimensions of reality, as seen in Section 7.1 (Fig. 1D₁) and Section 7.2 (Fig. 9F). These metrics mainly gauge similarity to real datasets without fully representing image fidelity. In our work, the generated objects in images occasionally do not match the actual scene in terms of target scale and brightness, which is an inevitable problem when generating large-scale image datasets. Such imperfections are largely due to the inherent limitations of current diffusion-based generative models, including GLIGEN. Achieving perfect photorealism, especially under open-world, complex driving conditions, remains a significant challenge for the field. A high frequency of such distorted images could introduce bias into the training dataset, leading to model overfitting on specific distortion patterns and subsequent underperformance in real-world scenarios. However, while this discrepancy is unavoidable, it only occurs with a low frequency and does not significantly impact the overall results in training models, as demonstrated in Section 7.3. Furthermore, within the sample view, these anomalous images are readily discernible from their surrounding counterparts, enabling users to efficiently

identify them and further exclude them from the dataset. Besides manually filtering out low-quality images, we can add more detailed constraints on objects’ depth and size, and use more advanced and stable image generation models like diffusion models for weather transformation in future work.

Striking a better trade-off of between image generation speed and reality. Evaluating the reality of generated images transcends subjective judgments, demanding more than just quantitative metrics, as the performance of generative models fluctuates across various scenarios, influencing the reality of the output. Despite the generative models mentioned in Section 5 holding promise for creating relatively realistic images, they face limitations in ensuring precise object insertion and rapid generation. Currently, if one wishes to generate a large batch of images in *HuGe*, time becomes a limiting factor. Moreover, to achieve faster generation speed, compromises often have to be made in terms of resolution and detail. In future work, we aim to use lightweight generative models to balance the generation speed and reality.

Enabling more fine-grained control of weather conditions. Enhancing precision in weather condition simulation is key to improving autonomous driving systems. Current models struggle with extreme or nuanced weather, particularly in replicating subtle lighting and visual effects. In the control panel (Fig. 1D₁), supported by an end-to-end generative algorithm, we currently allow only a single parameter to adjust each weather condition, which lack the conditions for fine-grained control of weather conditions. For instance, accurately generating images in complex weather conditions, such as rain or fog, requires more fine-grained control over elements like the intensity and distribution of water droplets or fog and their interactions with light and the environment. Furthermore, since generation and detection correspond to each other, it is only possible to detect weather deficiencies in one parameter, which limits our ability to identify imbalanced feature distributions in the dataset, as only one parameter of each weather condition does not allow for a detailed analysis of the dataset. In future work, we plan to integrate simulators like CARLA[12] to achieve more fine-grained weather control.

Scalability of knowledge-based object insertion. The knowledge-based controllable object insertion method discussed in Section 5.2 leverages spatial context learning, requiring user-input masks for CVAE processing a challenge for mask-lacking datasets like BDD100K. As an alternative, we employ optimization through random noise sampling, enhancing scalability and speed over CVAE, which is essential for rapid adaptation in diverse image sets. However, this method demands more from users in setting constraints, as it lacks the mask-based guidance of spatial context learning, highlighting a trade-off between scalability and user input reliance. Additionally, even when models are employed to learn object features and constraints are added to guide the output objects, it is still possible that some objects with an inappropriate scale are generated. Nonetheless, since these incongruent images are often very obvious, users can easily

filter out the flawed results. Therefore, our future work will explore developing intuitive algorithms to reduce reliance on detailed user inputs while enhancing the system's adaptability and fidelity for object insertion.

Other future work. Our approach inherently integrates human knowledge with generative AI, primarily focusing on testing and evaluating its usefulness and effectiveness in the field of autonomous driving. However, it can also be seamlessly adapted to anomaly detection scenarios with limited datasets, such as security screening, surveillance, and medical image analysis. For instance, in X-ray security screening for threat detection, this approach could digitally insert simulated threats like knives or guns into images, augmenting datasets to train more robust AI models. Moreover, with the rapid advancement of AI-driven video generation technologies, we plan to further extend our framework toward video data augmentation, incorporating temporal consistency and dynamic scene modeling to meet the growing demands of video-based applications. In addition, we will conduct a long-term study with more domain experts to further evaluate the usability and effectiveness of *HuGe*.

9. Conclusion

In this work, we designed and developed *HuGe*, a visual analytics system for generating autonomous driving images through a controllable, human-guided approach. It enables users to refine image datasets for enhanced training coverage. Our framework together with semi-automated generation methods streamline sample generation, reducing time and effort. Through case studies, a metric-based evaluation and expert interviews, we confirmed *HuGe*'s ability to effectively extend dataset coverage. Future efforts will focus on enhancing image quality with advanced techniques and further usability assessments with domain experts.

Acknowledgments

This work is supported by grants from the National Natural Science Foundation of China (No. 62302531) and the Science and Technology Planning Project of Guangdong Province (No. 2023B1212060029).

References

- [1] Abdal, R., Zhu, P., Mitra, N.J., Wonka, P., 2021. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)* 40, 1–21.
- [2] Ali, Y., Sharma, A., Haque, M.M., Zheng, Z., Saifuzzaman, M., 2020. The impact of the connected environment on driving behavior and safety: A driving simulator study. *Accident Analysis & Prevention* 144, 105643.
- [3] Anokhin, I., Solovev, P., Korzhenkov, D., Kharlamov, A., Khakhulin, T., Silvestrov, A., Nikolenko, S., Lempitsky, V., Sterkin, G., 2020. High-resolution daytime translation without domain labels, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7488–7497.
- [4] Bertucci, D., Hamid, M.M., Anand, Y., Ruangrotsakun, A., Tabatabai, D., Perez, M., Kahng, M., 2022. Dendromap: Visual exploration of large-scale image datasets for machine learning with treemaps. *IEEE Transactions on Visualization and Computer Graphics* 29, 320–330.
- [5] Bhat, S.F., Birk, R., Wofk, D., Wonka, P., Müller, M., 2023. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288* .
- [6] Cao, K., Liu, M., Su, H., Wu, J., Zhu, J., Liu, S., 2020. Analyzing the noise robustness of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics* 27, 3289–3304.
- [7] Chen, C., Guo, Y., Tian, F., Liu, S., Yang, W., Wang, Z., Wu, J., Su, H., Pfister, H., Liu, S., 2023. A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision. *IEEE Transactions on Visualization and Computer Graphics* .
- [8] Chen, Y., Rong, F., Duggal, S., Wang, S., Yan, X., Manivasagam, S., Xue, S., Yumer, E., Urtasun, R., 2021. Geosim: Realistic video simulation via geometry-aware composition for self-driving, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7230–7240.
- [9] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223.
- [10] DanielHfnr, 2023. Carla-Object-Detection-Dataset. URL: <https://github.com/DanielHfnr/Carla-Object-Detection-Dataset>.
- [11] Deng, Y., Zheng, X., Zhang, T., Liu, H., Lou, G., Kim, M., Chen, T.Y., 2022. A declarative metamorphic testing framework for autonomous driving. *IEEE Transactions on Software Engineering* .
- [12] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. Carla: An open urban driving simulator, in: *Proceedings of the Conference on Robot Learning*, PMLR. pp. 1–16.
- [13] Feng, Y., Wang, X., Wong, K.K., Wang, S., Lu, Y., Zhu, M., Wang, B., Chen, W., 2023. Promptmagician: Interactive prompt engineering for text-to-image creation. *IEEE Transactions on Visualization and Computer Graphics* .
- [14] Gao, R., Chen, K., Xie, E., Hong, L., Li, Z., Yeung, D.Y., Xu, Q., 2023. Magicdrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601* .
- [15] Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 1231–1237.
- [16] Gou, L., Zou, L., Li, N., Hofmann, M., Shekar, A.K., Wendt, A., Ren, L., 2020. Vatld: A visual analytics system to assess, understand and improve traffic light detection. *IEEE Transactions on Visualization and Computer Graphics* 27, 261–271.
- [17] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- [18] He, W., Zou, L., Shekar, A.K., Gou, L., Ren, L., 2021. Where can we help? a visual analytics approach to diagnosing and improving semantic segmentation of movable objects. *IEEE Transactions on Visualization and Computer Graphics* 28, 1040–1050.
- [19] Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G., 2023. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080* .
- [20] Hu, Y., Luo, C., Chen, Z., 2022. Make it move: controllable image-to-video generation with text descriptions, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18219–18228.
- [21] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P., 2017. Toward controlled generation of text, in: *Proceedings of International Conference on Machine Learning*, PMLR. pp. 1587–1596.
- [22] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134.
- [23] Jansen, P., Britten, J., Häusele, A., Segschneider, T., Colley, M., Rukzio, E., 2023. Autovis: Enabling mixed-immersive analysis of

Short Title of the Article

- automotive user interface interaction studies, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–23.
- [24] Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu, Z., Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., Jain, M., 2022. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. URL: <https://doi.org/10.5281/zenodo.7347926>, doi:10.5281/zenodo.7347926.
 - [25] Kenk, M.A., Hassaballah, M., 2020. Dawn: vehicle detection in adverse weather nature dataset. arXiv preprint arXiv:2008.05402 .
 - [26] Kynkääniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T., 2019. Improved precision and recall metric for assessing generative models. Advances in Neural Information Processing Systems 32.
 - [27] Li, B., Qi, X., Lukasiewicz, T., Torr, P., 2019a. Controllable text-to-image generation. Advances in Neural Information Processing Systems 32.
 - [28] Li, W., Pan, C., Zhang, R., Ren, J., Ma, Y., Fang, J., Yan, F., Geng, Q., Huang, X., Gong, H., et al., 2019b. Aads: Augmented autonomous driving simulation using data-driven algorithms. Science robotics 4, eaaw0863.
 - [29] Li, X., Bai, Y., Cai, P., Wen, L., Fu, D., Zhang, B., Yang, X., Cai, X., Ma, T., Guo, J., et al., 2023a. Towards knowledge-driven autonomous driving. arXiv preprint arXiv:2312.04316 .
 - [30] Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J., 2023b. Glichen: Open-set grounded text-to-image generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22511–22521.
 - [31] Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S., 2021. Editgan: High-precision semantic image editing. Advances in Neural Information Processing Systems 34, 16331–16345.
 - [32] Liu, J., Lu, B., Xiong, M., Zhang, T., Xiong, H., 2023. Adversarial attack with raindrops. arXiv preprint arXiv:2302.14267 .
 - [33] Menze, M., Geiger, A., 2015. Object scene flow for autonomous vehicles, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3061–3070.
 - [34] Mohammed, A.S., Amamou, A., Ayevide, F.K., Kelouwani, S., Agbossou, K., Zioui, N., 2020. The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review. Sensors 20, 6532.
 - [35] Muhammad, K., Ullah, A., Lloret, J., Del Ser, J., de Albuquerque, V.H.C., 2020. Deep learning for safe autonomous driving: Current challenges and future directions. IEEE Transactions on Intelligent Transportation Systems 22, 4316–4336.
 - [36] Muşat, V., Fursa, I., Newman, P., Cuzzolin, F., Bradley, A., 2021. Multi-weather city: Adverse weather stacking for autonomous driving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2906–2915.
 - [37] Pei, K., Cao, Y., Yang, J., Jana, S., 2017. Deepxplore: Automated whitebox testing of deep learning systems, in: Proceedings of the 26th Symposium on Operating Systems Principles, pp. 1–18.
 - [38] Powell, M.J., 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. Springer.
 - [39] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M., 2022. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 1, 3.
 - [40] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems 28.
 - [41] Sakaridis, C., Dai, D., Van Gool, L., 2021. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10765–10775.
 - [42] Sohn, K., Lee, H., Yan, X., 2015. Learning structured output representation using deep conditional generative models. Advances in Neural Information Processing Systems 28.
 - [43] Song, Z., He, Z., Li, X., Ma, Q., Ming, R., Mao, Z., Pei, H., Peng, L., Hu, J., Yao, D., et al., 2023. Synthetic datasets for autonomous driving: A survey. IEEE Transactions on Intelligent Vehicles .
 - [44] Tian, Y., Pei, K., Jana, S., Ray, B., 2018. Deepfest: Automated testing of deep-neural-network-driven autonomous cars, in: Proceedings of the 40th International Conference on Software Engineering, pp. 303–314.
 - [45] Tremblay, M., Halder, S.S., De Charette, R., Lalonde, J.F., 2021. Rain rendering for evaluating and improving robustness to bad weather. International Journal of Computer Vision 129, 341–360.
 - [46] Wang, Q., L'Yi, S., Gehlenborg, N., 2023. Drava: Aligning human concepts with machine learning latent dimensions for the visual exploration of small multiples, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–15.
 - [47] Wen, M., Park, J., Cho, K., 2020. A scenario generation pipeline for autonomous vehicle simulators. Human-centric Computing and Information Sciences 10, 1–15.
 - [48] Wu, H., Yunas, S., Rowlands, S., Ruan, W., Wahlström, J., 2023. Adversarial driving: Attacking end-to-end autonomous driving, in: Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), IEEE, pp. 1–7.
 - [49] Xie, X., Cai, X., Zhou, J., Cao, N., Wu, Y., 2018. A semantic-based method for visualizing large image collections. IEEE Transactions on Visualization and Computer Graphics 25, 2362–2377.
 - [50] Xu, W., Souly, N., Brahma, P.P., 2021. Reliability of gan generated data to train and validate perception systems for autonomous vehicles, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 171–180.
 - [51] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T., 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2636–2645.
 - [52] Zhang, L., Rao, A., Agrawala, M., 2023. Adding conditional control to text-to-image diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847.
 - [53] Zhang, M., Zhang, Y., Zhang, L., Liu, C., Khurshid, S., 2018. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems, in: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 132–142.
 - [54] Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232.

Ethical Approval

This study does not contain any studies with Human or animal subjects performed by any of the authors.

Declaration of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.