DriveDiTFit: Fine-tuning Diffusion Transformers for Autonomous Driving Data Generation

JIAHANG TU, Zhejiang University, Hangzhou, China
WEI JI, National University of Singapore, Singapore, Singapore
HANBIN ZHAO and CHAO ZHANG, Zhejiang University, Hangzhou, China
ROGER ZIMMERMANN, National University of Singapore, Singapore, Singapore
HUI QIAN, Zhejiang University, Hangzhou, China

In autonomous driving, deep models have shown remarkable performance across various visual perception tasks with the demand of high-quality and huge-diversity training datasets. Such datasets are expected to cover various driving scenarios with adverse weather, lighting conditions, and diverse moving objects. However, manually collecting these data presents huge challenges and is expensive. With the rapid development of large generative models, we propose DriveDiTFit, a novel method for efficiently generating autonomous *Driv*ing data by *Fi*ne-*t*uning pre-trained *Di*ffusion *T*ransformers (DiTs). Specifically, DriveDiTFit utilizes a gap-driven modulation technique to carefully select and efficiently fine-tune a few parameters in DiTs according to the discrepancy between the pre-trained source data and the target driving data. Additionally, DriveDiTFit develops an effective weather and lighting condition embedding module to ensure diversity in the generated data, which is initialized by a nearest-semantic-similarity initialization approach. Through progressive tuning scheme to refine the process of detail generation in early diffusion process and enlarging the weights corresponding to small objects in training loss, DriveDiTFit ensures high-quality generation of small moving objects in the generated data. Extensive experiments conducted on driving datasets confirm that our method could efficiently produce diverse real driving data.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Computer vision**;

Additional Key Words and Phrases: Diffusion Transformer, Driving Image Generation, Fine-tuning

ACM Reference format:
Jiahang Tu, Wei Ji, Hanbin Zhao, Chao Zhang, Roger Zimmermann, and Hui Qian. 2025. DriveDiTFit: Fine-tuning Diffusion Transformers for Autonomous Driving Data Generation. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 3, Article 85 (February 2025), 29 pages.
https://doi.org/10.1145/3712064

Code is available at: https://github.com/TtuHamg/DriveDiTFit
This work was supported in part by the National Natural Science Foundation of China under Grant 62402430, 62206248, and Aeronautical Science Foundation of China 20240048076001.
Authors' Contact Information: Jiahang Tu, Zhejiang University, Hangzhou, China; e-mail: tujiahang@zju.edu.cn; Wei Ji, National University of Singapore, Singapore, Singapore; e-mail: weiji0523@gmail.com; Hanbin Zhao (corresponding author), Zhejiang University, Hangzhou, China; e-mail: zhaohanbin@zju.edu.cn; Chao Zhang, Zhejiang University, Hangzhou, China; e-mail: zczju@zju.edu.cn; Roger Zimmermann, National University of Singapore, Singapore, Singapore; e-mail: rogerz@comp.nus.edu.sg; Hui Qian, Zhejiang University, Hangzhou, China; e-mail: qianhui@zju.edu.cn.

ACM 1551-6865/2025/2-ART85
https://doi.org/10.1145/3712064

1 Introduction

Recent years have witnessed a rapid development of deep learning models on the autonomous driving application [5, 41]. The performance of data-driven deep models often corresponds to the quality and diversity of the driving data, which necessitates the construction of high-quality and huge-diversity training datasets [1, 8, 52]. Specifically, such datasets are expected to cover various driving scenarios with adverse weather, lighting conditions, and diverse moving objects. As demonstrated in Figure 1, in addition to sunny and day scenarios, driving datasets should contain snowy, rainy, and night scenarios with vehicles in the field of view. However, manually collecting these data is challenging and expensive, especially for data containing extreme weather scenarios and small moving objects. Due to the outstanding capabilities of **Large-Scale Generative Models (LGM)** [10, 33], this article focuses on generating data efficiently with LGM to further promote the development of deep models on autonomous driving.

Directly training an LGM from scratch for driving scenarios is time-consuming and resource-expensive, some of recent works try to fine-tune a pre-trained LGM on specific downstream tasks. For example, Xie et al. [51] propose to fine-tune biases in **Diffusion Transformers (DiTs)** and validate the efficiency on commonly used fine-grained natural classification datasets. Hence, we focus on fine-tuning a pre-trained LGM efficiently to adapt to the complex autonomous driving scenarios. The LGM is always pre-trained on a natural classification dataset (e.g., ImageNet [6]) and encodes the class-specific knowledge with a class embedding module. As depicted in Figure 1, such dataset mainly includes a prominent natural-category object located in the central position of the image, but driving data usually contain vehicles with diverse weather and lighting conditions. Motivated by these observations, we consider the discrepancy and aim to design a parameter-efficient LGM fine-tuning method tailored for autonomous driving scenarios.

In this article, we propose DriveDiTFit to efficiently generate high-quality and huge-diversity autonomous *Driv*ing data by *Fi*ne-*t*uning pre-trained *Di*ffusion *T*ransformers. Specifically, DriveDiTFit utilizes a gap-driven modulation scheme to fine-tune only a few parameters of the pre-trained DiTs. To generate adverse weather and lighting conditions, DriveDiTFit effectively embeds the weather- and lighting condition-specific knowledge by an expandable condition embedding module, which is initialized by a nearest-semantic-similarity initialization approach. Besides, we observed that the noise schedule in the diffusion process [19, 29] can greatly affect the quality of objects in the generated data, as depicted in Figure 2. Our DriveDiTFit designs a **Spoon-Cosine (Scos)** noise schedule and progressively adjusts the noise intensity to mitigate the impact on original parameters. To further ensure the generated quality of small objects, we explicitly introduce the extra position knowledge of small objects and develop an **Object-Sensitive Loss (OSL)** function by enlarging the weights corresponding to the regions of the small objects. Overall, the contribution of our work is threefold:

- We propose a gap-driven modulation technique for parameter-efficient fine-tuning DiT models to address the large gap between source datasets and target datasets.
- We present a semantically relevant embedding initialization method, leveraging the prior knowledge to embed the weather and lighting conditions. This approach enhances tuning convergence and improves the generation quality.
- To enhance the detail fidelity of generated small objects in driving scenarios, we implement a progressive tuning scheme with an innovative Scos noise schedule and introduce a loss function sensitive to small objects.

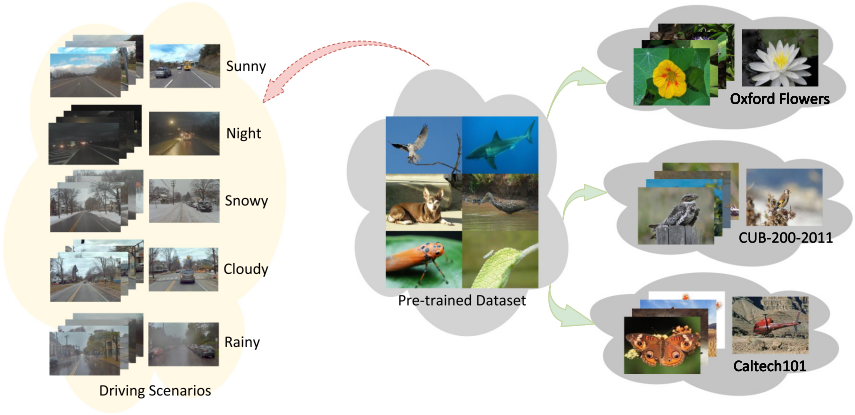

Fig. 1. There is an apparent discrepancy between pre-trained datasets and driving scenario datasets. Pre-trained datasets usually feature certain categories of objects prominently displayed within the images, which is similar to the fine-grained classification datasets, such as CUB-200-2011 and Oxford Flowers. However, driving scenario datasets are more complex and contain multiple objects, including roads, vehicles, and buildings, with diverse weather and lighting conditions.

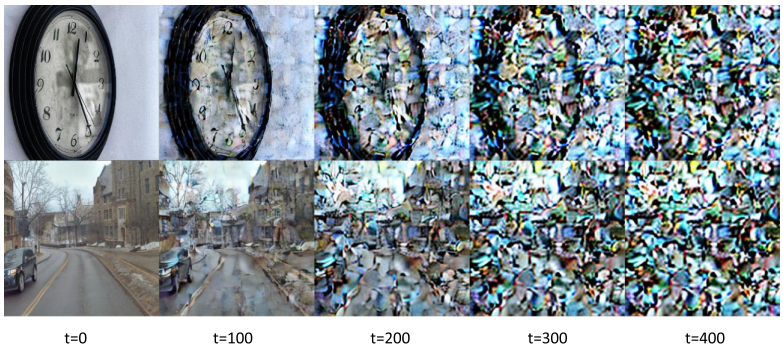

Fig. 2. The object information can be generated sufficiently in the denoising process [19], when it loses slowly in the diffusion process. The conventional noise schedule [31] makes big objects in classification dataset lose slowly (top row, clock, t from 0 to 400), whereas it causes smaller objects in driving data to fade more rapidly (bottom row, vehicles, t from 0 to 200). An appropriate noise schedule is necessary for driving data generation.

2 Related Work

2.1 Autonomous Driving Datasets

The KITTI dataset [13], widely utilized in research on driving algorithms, serves as a benchmark for tasks like 2D object detection and optical flow. It features a rich collection of RGB, LiDAR, and GPS/IMU data, albeit predominantly under daytime and clear weather conditions. To broaden the scope of environmental conditions, the Waymo [40] and Zenseact [1] datasets cover night scenarios and introduce dawn and dusk scenarios, respectively, and the nuScenes [4] dataset expands coverage to rainy weather but lacks snow scenes. Notably, BDD100K [52] dataset collects five types of weather conditions, including clear, cloudy, rainy, snowy, and foggy settings. As shown in Table 1, given the unbalanced diversity in conditions, the predominance of clear weather and daylight scenarios in these datasets often leads to decreased performance of visual models under nighttime or adverse weather conditions. Ithaca365 [8] collects repeated trajectories under

Table 1. Statistical Comparisons of Weather and Lighting Conditions in Driving Datasets

Driving datasets	Weather Conditions (%)				Time of Day (%)		
	Clear/Cloudy	Rainy	Snowy	Foggy	Day	Night	Dawn/Dusk
KITTI [13]	100	-	-	-	100	-	
nuScenes [4]	78.1	21.9	-	-	88.4	11.6	-
Waymo [40]	99.4	0.6	-	-	81.0	9.9	9.1
Zenseact [1]	80.2	15.7	2.0	2.1	77.3	19.0	3.6
BDD100K [52]	66.0	6.9	7.8	0.1	52.6	40.4	7.2

diverse scene, weather, time, and traffic conditions over 1.5-year period, but its data scale is smaller compared to the previously mentioned datasets. The manual acquisition of driving datasets from real-world scenarios is both time-consuming and labor-intensive.

2.2 Diffusion Models

Diffusion Probabilistic Models (DPMs) [18, 38] have emerged as a powerful class of generative models, surpassing generative adversarial networks [14] and variational autoencoders [23, 47] in a variety of tasks, including text-to-image generation [35, 42, 50, 54], image editing [26, 46], and video synthesis [3, 17, 45]. The essence of DPMs lies in incrementally mapping Gaussian noise to intricate distributions related to datasets. Given a certain noise schedule, the diffusion phase converts a data distribution to a standard Gaussian distribution by adding noise. The denoising phase primarily adopts a UNet [36] architecture to iteratively reverse the noise addition, thereby reconstructing the original data distribution. Inspired by breakthroughs in natural language processing [7] and vision transformer
[9, 11], architectures based on transformers [44] have been proposed and replaced the UNet backbone, achieving state-of-the-art results [2, 31] on benchmark datasets, such as CIFAR10 [24] and ImageNet.

2.3 Fine-tuning for Diffusion Models

The emergence of numerous high-quality diffusion models has drawn the attention of researchers, prompting them to explore fine-tuning approaches tailored to their specific requirements. Techniques like Textual Inversion[12] and Dreambooth [37] introduce novel token identifiers for personalized text embedding adjustments, although their application remains largely within text-to-image paradigms and customized datasets are similar to source datasets. DiffFit [51] proposed a parameter-efficient method by fine-tuning biases and scale factors in DiTs. Moon et al. [27] explore the integration of time-fusion adapters within attention mechanisms on limited datasets. **Visual Prompt Tuning (VPT)** [22] freezes transformer blocks and inserts a few learnable prompt embeddings in different layers for downstream tasks tuning. Nevertheless, these works predominantly focus on classification datasets, with limited exploration in the efficiency on scenario datasets. Moreover, the interaction between the original noise schedule and the target dataset characteristics remains an underexplored area.

3 Methods

Our goal is to fine-tune the pre-trained DiTs from classification datasets to driving datasets. To achieve this, we propose the DriveDiTFit framework that utilizes gap-driven modulation techniques to fine-tune the condition **Multilayer Perceptron (MLP)** and attention blocks and employs **Semantic-Selective Embedding Initialization (SSEI)** to accelerate fine-tuning convergence.

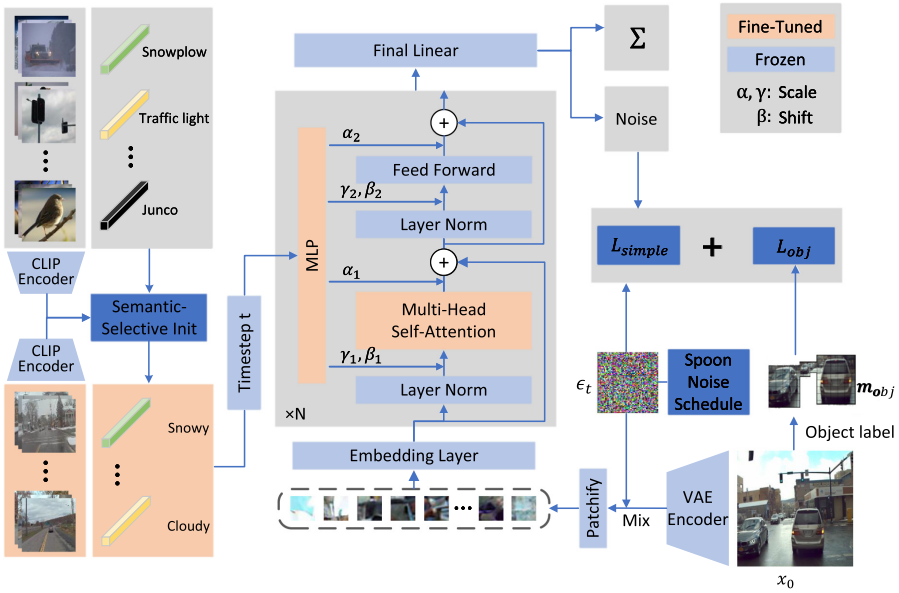

Fig. 3. Our framework for diverse driving scenario generation consists of three key components—(i) gap-driven modulation techniques on the condition MLP and attention blocks (Section 3.2); (ii) accelerating convergence and enhancing quality by initiating with high semantic similarity embeddings via a CLIP encoder (Section 3.3); (iii) adopting progressive tuning scheme with novel Scos noise schedule (Section 3.4.1) and applying vehicle bounding box masks on training loss (Section 3.4.2) for precise object representation.

Additionally, DriveDiTFit adopts a progressive tuning scheme with the novel Scos noise schedule and applies vehicle bounding box masks to the training loss for precise object representation. Figure 3 illustrates our complete process.

3.1 Preliminaries

Before introducing our novel fine-tuning method for DPMs, we briefly revisit the foundational principles of DPMs. Given a data distribution $x_0 \sim q_{data}(x)$, the diffusion phase iteratively adds noise ϵ_t to the sample x_t until x_T, following a certain noise schedule and time t. This process can be described as follows:

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

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{2}$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Here, β_t represents the noise intensity at each step, and a large number of steps (T) enables x_T to approximate a Gaussian distribution closely.

The essence of DPMs lies in their ability to reverse this noise addition process, effectively generating samples from the original data distribution by learning a sequence of reverse mappings. They aim to learn the inverse process $p_\theta(x_{t-1}|x_t)$ and approximate the posterior $q(x_{t-1}|x_t, x_0)$, which are defined as follows:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{4}$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathbf{I}), \tag{5}$$

where $\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t$ and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. The hypothesis by Ho et al. [18] Σ_θ is not learnable and reparameterize the $\mu_\theta(x_t, t)$ through Equations (3) and (5):

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right). \tag{6}$$

The simple loss function can be written as follows:

$$\mathcal{L}_{\text{simple}} = E_{t, x_0, \epsilon_t} \left[||\epsilon_t - \epsilon_\theta(x_t, t)||^2 \right]. \tag{7}$$

During inference, DPMs generate new samples by first initializing x_T from the Gaussian distribution. The model then sequentially calculates x_{t-1} through Equation (4) until reaching x_0, thus reconstructing or generating new data points that mimic the original data distribution.

3.2 Architecture Fine-tuning

As previously discussed, driving scenarios present a distinct challenge from traditional classification generation. Specifically, DiTs in driving scenarios must generate images across varied weather and lighting conditions while creating fine details of small objects. This requirement underscores a significant gap: mastering single-category object is insufficient. Instead, models must grasp the global weather and lighting style and positional relationships with multiple objects. This complexity suggests that methods solely fine-tuning biases and scale factors, as proposed in DiffFit [51], may fall short for driving datasets. Our hypothesis is that the gaps between original datasets and fine-tuned datasets influence the number of parameters that need to be adjusted. Inspired by AdaLN [21], which leverages style statistics information to modulate input images and overcomes the gaps between various art styles, we utilize the modulation approach to bridge the gaps between the pre-trained dataset and driving datasets. Specifically, we select different modules in DiTs according to the significant gap. As the changes in conditions and different layout demands of driving datasets, we modulate the weights within the condition MLP and the **Multi-Head Self-Attention (MHSA)** block, employing **Low-Rank Adaptation (LoRA)** to enhance training efficiency. We obtain the modulated weight $\tilde{W} \in \mathbb{R}^{d_{out} \times d_{in}}$ like:

$$\tilde{W} = W \odot \Gamma + B, \tag{8}$$

where $\Gamma \in \mathbb{R}^{d_{out} \times d_{in}}$ and $B \in \mathbb{R}^{d_{out} \times d_{in}}$ present the fine-tuning parameters. These can be effectively represented by two low-rank matrices:

$$\Gamma = \Gamma^{out} \otimes \Gamma^{in}, B = B^{out} \otimes B^{in}, \tag{9}$$

with $\Gamma^{out}, B^{out} \in \mathbb{R}^{d_{out} \times r}$ and $\Gamma^{in}, B^{in} \in \mathbb{R}^{r \times d_{in}}$. Detailed experiments demonstrate the validity of our fine-tuning scheme on the structure. Furthermore, the generation of local objects in driving scenarios should have a narrower focus compared to category objects in classification datasets. We adopt an approach similar to Peebles and Xie [31] and introduce 2D **Rotary Positional Embeddings (RoPE)** [39], a technique prevalent in natural language processing, to emphasize the generation of local objects. For an input $z \in \mathbb{R}^{c \times h \times w}$ in the latent space, we can obtain $\frac{H}{p} \times \frac{W}{p}$ $u \in \mathbb{R}^{c \times p \times p}$ tokens. Applying RoPE along each spatial axis and concatenating, we obtain the position embedding $P_{i,j} \in \mathbb{R}^{d \times d}$:

$$P_{i,j} = \begin{bmatrix} RoPE(i) & 0 \\ 0 & RoPE(j) \end{bmatrix}, \tag{10}$$

where RoPE$(\cdot) \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}$ and d represents hidden channel number. The local attention mechanism can be written as:

$$(P_{i,j}q_{i,j})^T(P_{i',j'}k_{i',j'}) = q_{i,j}^T P_{i-i',j-j'}k_{i',j'}. \tag{11}$$

As $P_{i-i',j-j'}$ is an orthogonal constant matrix, it preserves the magnitude of vectors without adding new learnable parameters to the model, thus ensuring efficiency and model simplicity.

3.3 SSEI

The DiT model incorporates learnable embeddings that encapsulate categorical information from classification datasets. Hence, we develop a weather and lighting condition embedding to control diverse scenario generation. Directly initializing the embeddings from scratch is deemed impractical due to the significant impact on parameters. Traditional methods of initializing embeddings through random selection of pre-trained embeddings introduce a certain degree of variability. To address this challenge, we propose an SSEI approach. We employ a pre-trained CLIP encoder [32] to extract semantic features $\{z_i^s\}_{i=1}^N$ and $\{z_j^{ad}\}_{j=1}^M$ from both pre-trained datasets and driving datasets. We then measure the semantic similarity between z_i^s and z_j^{ad} with cosine similarity to find an appropriate class embedding $c_{i^*}^s$ for certain scenario condition c_j^{ad}. The nearest-semantic-similarity embedding initialization can be written as follows:

$$i^* = \underset{i}{\operatorname{argmax}} \frac{z_i^s \cdot z_j^{ad}}{\|z_i^s\|_2 \cdot \|z_j^{ad}\|_2}. \tag{12}$$

Our hypothesis is that image features closely aligned in semantic space are likely to have similar information in DiT embeddings and also exhibit similar distribution characteristics. Through ablation studies, we have demonstrated that our SSEI approach significantly accelerates model convergence and yields superior performance outcomes. This method effectively leverages semantic correlations to enhance the initialization process, setting a robust foundation for better learning and adaptation in driving scenario modeling.

3.4 Progressive Tuning Scheme

3.4.1 Scos Noise Schedule. The conventional linear noise schedule approach, typically employed for classification datasets, proves inadequate for driving datasets. This insight draws inspiration from the work of Hoogeboom et al. [19] on high-resolution images synthesis, which revealed that traditional noise schedules only afford a small time window to decide the global structure of the image at the late diffusion process, retaining good visual quality of high-resolution generation. In contrast, as we demonstrate in Figure 2, small objects within scenarios tend to be submerged in the early diffusion process, which lead to a shorter time window for the reverse process of small objects generation. The underlying cause for this phenomenon is that at a resolution of 256 × 256 pixels, small objects in driving datasets are described by fewer pixels compared to those in classification datasets. Consequently, at equivalent noise intensity, small objects become harder to identify.

A naive idea is to alter the original schedule during the fine-tuning, such as employing cosine schedule, which is notably beneficial for driving datasets as it reduces the noise intensity in the early stages of diffusion. However, the practical application of this seemingly straightforward adjustment during fine-tuning presents challenges as it causes great damage to the learned mapping from the Gaussian distribution to the dataset distribution. Inspired by squared cosine noise schedule of Nichol and Dhariwal [29], we discover that adjusting the power term s effectively captures the

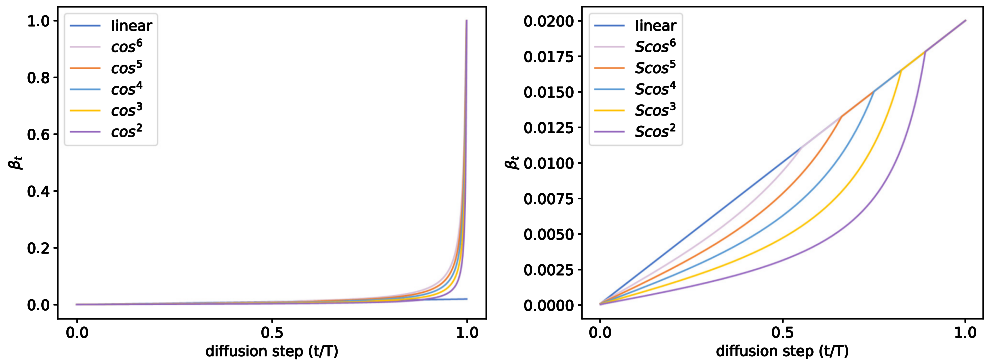

Fig. 4. In the diffusion process, β_t varies between the cosine noise schedule proposed by Nichol and Dhariwal and Scos noise schedule.

gradual transition from a linear to a cosine schedule:

$$f_s(t) = \cos^s\left(\frac{t/T + b}{1 + b} \cdot \frac{\pi}{2}\right), \tag{13}$$

$$\bar{\alpha}_t = \frac{f_s(t)}{f_s(0)}. \tag{14}$$

Nevertheless, as demonstrated in Figure 4, we observe a significant change in the β_t value, which represents noise intensity, at the endpoint of the diffusion process. This change obviously differs from the diffusion patterns learned by pre-trained models. This discovery suggests that the traditional fine-tuning methods on model architecture may not adequately adapt to such changes. Consequently, we comprehensively consider the advantages and disadvantages of the squared cosine noise schedule and propose a novel noise schedule to optimize the adjustment of noise intensity throughout the diffusion process, which provides a novel perspective for fine-tuning pre-trained diffusion models. We have termed this strategy the "spoon-cosine" (Scos) noise schedule due to its unique β_t value change curve, which resembles the shape of a spoon. Specifically, during the early stages of the diffusion process, we employ a cosine noise schedule with lower noise intensity to delay the complete loss of fine details and extend the time window for detail generation in the denoising process. At the intersection of the cosine and linear noise intensity curves, we transition to the linear noise schedule with a lower noise intensity to avoid abrupt changes in noise levels, which could potentially disrupt the model parameters. Thus, the Scos noise schedule can be formulated as follows:

$$Scos^s(t) = \min\left(1 - \frac{f_s(t)}{f_s(t-1)}, \frac{\beta_T - \beta_0}{T}t + \beta_0\right). \tag{15}$$

The Scos noise schedule can gradually adjust the exponent term s, allowing for a progressive reduction in noise intensity during the early stages of the diffusion process.

In the progressive fine-tuning strategy, we first adjust the model from the pre-trained dataset to the driving dataset using the original linear noise schedule on the driving dataset. In Figure 5, we replace the linear noise schedule with a Scos^6 schedule, and at specified training intervals τ, gradually transition from Scos^6 to Scos^2 noise schedule, facilitating the model's ability to generate more intricate and detailed representations.

3.4.2 OSL. For the generation of high-quality objects, particularly of vehicles, our work proposes an OSL designed to enhance the precision and quality of generated objects. Specifically, we

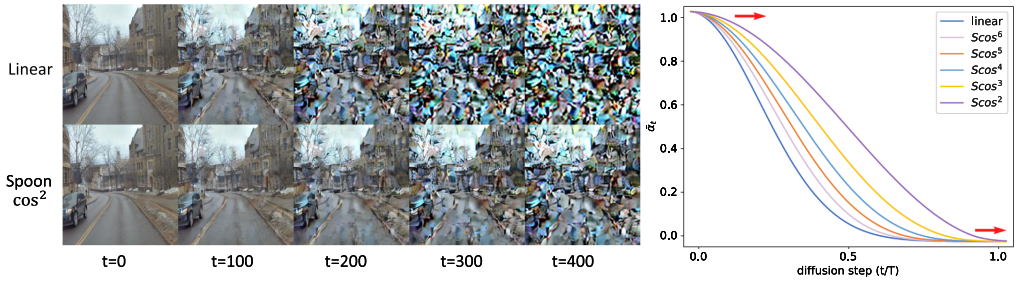

Fig. 5. Left: The linear noise schedule (top row) and Scos^2 noise schedule (bottom row) are adopted on an driving scenario sample. Right: $\bar{\alpha}_t$ in diffusion process for the linear noise schedule and Scos noise schedule with different powers.

introduce additional supervision signals into the original loss function, leveraging the bounding box information available within driving datasets. More precisely, we select bounding boxes associated with vehicles and incorporate these prior masks m_{obj} into the loss function by enlarging the weights corresponding to certain regions, which encourages the model to allocate more focus in the generation of vehicles. This OSL is defined as:

$$\mathcal{L}_{obj} = E_{t,x_0,\epsilon_t}\left[||m_{obj}(\epsilon_t - \epsilon_\theta(x_t, t))||^2\right]. \tag{16}$$

The final loss of DriveDiTFit is the combination of the simple denoising loss and OSL:

$$\mathcal{L} = \mathcal{L}_{\text{simple}} + \lambda_o \mathcal{L}_{\text{obj}}, \tag{17}$$

where λ_o is the coefficient of the OSL.

4 Experiments

In this section, we present experiments to validate the effectiveness and motivation of our proposed method. Quantitative and qualitative analyses present that DriveDiTFit can generate high-quality and huge-diversity driving data.

4.1 Implementation

Datasets. Ithaca365 [8] is an autonomous driving dataset recorded along a 15-km route under diverse weather (clear, cloudy, rainy, snowy) and lighting (day, night) conditions. It provides scenario image data with balanced labels across varying scenarios, including urban, highway, and rural environments. BDD100K [52] is another substantial driving dataset renowned for its size and diversity. It comprises 70,000 images within the training dataset and covers a small amount of foggy scenario data, alongside the aforementioned weather and lighting conditions. nuScenes [4] is a large-scale multimodal dataset for autonomous driving, designed to provide a comprehensive view of the entire sensor suite. The dataset consists of 1,000 scenes, each 20 seconds long, which comprises 1.4 million driving images.

Metrics. We assess the effectiveness of DriveDiTFit in terms of sample quality and the coverage of the data manifold. **Fréchet Inception Distance (FID)** [16] leverages pooling features from Inception-V3 to calculate the Kullback-Leibler divergence between real and generated samples. **Spatial Fréchet Inception Distance (sFID)** [28] is a new version of FID that replaces traditional pooling features with spatial features. sFID is better at capturing the spatial relationships between images, making it particularly suitable for images with complex structures. This feature allows sFID to more accurately reflect the high-level structure of images, especially when handling vehicle and environmental details in driving data. Besides, we utilize improved precision and recall

metrics [25] to evaluate the fidelity and diversity of samples, respectively. Precision measures the proportion of generated samples that are close to the real-data distribution, focusing on whether the generated samples accurately represent the vehicles, environment, and other details in driving scenes. Recall measures the proportion of real data covered by the generated samples, emphasizing whether the generative model can capture the diversity of driving datasets, i.e., various weather and lighting environments. Additionally, we select the object detection downstream task to validate the performance improvement of perception models using the generated dataset. We employ the **Mean Average Precision (mAP)** [34] as the evaluation metric, which represents the average area under the precision–recall curve for all classes in the images.

Details. We choose DiT-XL/2 as our base model, which are pre-trained on ImageNet at 256×256 resolution over 7 million steps. To maintain consistency with the pre-trained models, the driving dataset is resized to 256×256 resolution. For the Ithaca365 dataset, we randomly select samples from sunny, cloudy, rainy, snowy, and night conditions, totaling 6,000 evaluation samples. For the BDD100K dataset, we select samples from sunny, cloudy, overcast, rainy, snowy, and night conditions, totaling 30,000 samples. We employ a similar approach to BDD100K to evaluate the nuScenes dataset. We use the denoising diffusion implicit models sampler with 250 sampling steps, generating images with a resolution of 256×256.

We select DiffFit as our primary comparison method since it has been tested with DiT-XL/2 on a broad range of downstream classification datasets. Following the setup of DiffFit, we fine-tune the DiT model by adjusting the QKV-Linear and output layer parameters in layers 1–14 with a learning rate of $1e^{-3}$. Additionally, we also re-implement several fine-tuning methods, including BitFit [53], Time-Adapter [27], VPT [22], and LoRA [20]. For BitFit, we use the same configuration as DiffFit. For Time-Adapter, we adapt the Time-Adapter method to the DiT blocks by inserting the adapter module after the self-attention layer and set the learning rate to $2e^{-4}$, following the original paper's settings. For VPT, we find that VPT is sensitive to both depth and token count, and training is highly unstable. Therefore, we choose a stable configuration suitable for the driving dataset: we insert a prompt consisting of three tokens in layers 1–14, using the same $1e^{-3}$ learning rate to fine-tune the DiT model. For LoRA, we use the default configuration and select rank = 4 to fine-tune the attention module of DiT. We use four A800 GPUs with a global batch size of 256 for 500 iterations, with a fixed random seed of 0. We select ViT-L-14 as the semantic encoder and randomly choose 100 samples from each class and compute the average as the cosine similarity measure between two classes. τ is set to the number of training steps.

For DriveDiTFit, we fine-tune the conditional MLP layers and the QKV-Linear in the self-attention layers within the DiT blocks from layers 1 to 28, setting the fine-tuning learning rate to $1e^{-3}$ and the rank to 4. The two low-rank matrices are initialized using the Kaiming initialization and zero initialization, respectively. For the noise schedule hyperparameters, we use the default settings: $\beta_0 = 1e^{-4}$, $\beta_T = 2e^{-2}$, and $s = 2$.

4.2 Quantitative Evaluation

Comparison with the State-of-the-Arts. In this section, we show the quantitative comparison of our proposed DriveDiTFit with other fine-tuning methods. As shown in Table 2, under the condition of an acceptable amount of fine-tuning parameters and comparable other metrics, our proposed DriveDiTFit outperforms DiffFit and BitFit in terms of the FID and precision metric. This corroborates our initial motivation, suggesting that for fine-tuning tasks with the significant gap in datasets, merely adjusting biases and scaling factors are insufficient. Due to the small size of Ithaca365, tuning more parameters can lead to overfitting, resulting in decreased performance on the Full Fine-tuning and Time-Adapter method. The generation samples of different methods are included in Appendix A.

Table 2. The Performance Comparison of DriveDiTFit and Other Fine-Tuning Methods on the Ithaca365 Dataset

Method	FID↓	sFID↓	Precision↑	Recall ↑	Params (M)
Full Fine-tune	37.98	31.95	0.571	0.648	675.1 (100%)
Time-Adapter [27]	30.82	27.41	0.558	0.711	260.3 (38.5%)
LoRA-R4 [20]	37.38	34.05	0.425	0.649	3.000 (0.44%)
VPT-Deep [22]	35.32	31.24	0.358	0.677	1.380 (0.20%)
BitFiT [53]	29.23	31.40	0.379	0.751	0.490 (0.07%)
DiffFit [51]	27.98	31.11	0.386	0.773	0.590 (0.09%)
DriveDiTFit	18.64	26.13	0.626	0.795	2.372 (0.35%)

Our proposed approach achieves the best generative performance while only requiring fine-tuning 0.35% of the parameters. The underline indicates the best result.

Table 3. The Performance Comparison of DriveDiTFit and Other Fine-Tuning Methods on the BDD100K Dataset

Method	FID↓	sFID↓	Precision↑	Recall ↑	Params (M)
Full Fine-tune	38.96	12.70	0.531	0.610	675.1 (100%)
Time-Adapter	45.65	15.25	0.495	0.576	260.3 (38.5%)
LoRA-R4	44.35	15.18	0.651	0.578	3.000 (0.44%)
VPT-Deep	35.32	12.32	0.547	0.573	1.380 (0.20%)
BitFiT	42.18	12.96	0.631	0.603	0.490 (0.07%)
DiffFit	46.57	15.06	0.651	0.585	0.590 (0.09%)
DriveDiTFit	33.38	11.93	0.659	0.611	2.372 (0.35%)

The underline indicates the best result.

We conduct additional experiments on the BDD100K and nuScenes datasets to validate the effectiveness of our proposed DriveDiTFit on large-scale driving datasets. In Tables 3 and 4, our method outperforms BitFit and DiffFit by a margin on FID and sFID and achieves slightly better performance on precision and recall. These results suggest that fine-tuning only a small number of bias parameters is insufficient to bridge the gap between the original dataset and the driving dataset. We observe that the evaluation results of Time-Adapter and VPT-Deep on the BDD100K and nuScenes datasets lack consistency. For instance, VPT-Deep achieves a strong FID on the BDD100K dataset compared to other traditional fine-tuning methods; however, its performance on the nuScenes dataset is significantly lower than that of other methods. We attribute this discrepancy to the parameters introduced by VPT-Deep and Time-Adapter, which cannot be effectively initialized using zero or identity initialization, leading to instability during training. Compared to Full Fine-tune, Time-Adapter, and LoRA-4 that have higher parameter costs, our method achieves superior evaluation results across all four evaluation metrics, which demonstrates that our method strikes a good tradeoff between generative capabilities and parameter cost.

Impact of DiT's Modules. We first explore the effects of fine-tuning various modules of DiTs on their generative performance. As seen in Table 5, fine-tuning the MHSA modules and the condition MLP (MLP_c) significantly enhances the performance. This may stem from the complex information in driving conditions affecting the attention mechanisms. Moreover, since the DriveDiTFit method

Table 4. The Performance Comparison of DriveDiTFit and Other Fine-Tuning Methods on the nuScenes Dataset

Method	FID↓	sFID↓	Precision↑	Recall ↑	Params (M)
Full Fine-tune	34.58	27.44	0.585	0.463	675.1 (100%)
Time-Adapter	39.04	27.73	0.589	0.420	260.3 (38.5%)
LoRA-R4	38.56	27.59	0.580	0.433	3.000 (0.44%)
VPT-Deep	51.06	32.85	0.412	0.325	1.380 (0.20%)
BitFiT	39.96	27.72	0.551	0.434	0.490 (0.07%)
DiffFit	37.47	27.40	0.563	0.458	0.590 (0.09%)
DriveDiTFit	32.49	27.17	0.596	0.478	2.372 (0.35%)

The underline indicates the best result.

Table 5. Ablation Experiments on Different Modules of DiTs on the Ithaca365 Dataset

MHSA	MLP_b	MLP_c	Patch Conv	FID↓	Params (M)
✓				30.89	1.081 (0.16%)
✓		✓		27.45	2.372 (0.35%)
✓	✓	✓		27.09	3.984 (0.58%)
✓	✓	✓	✓	29.84	3.991 (0.59%)

Our proposed DriveDiTFit selects MHSA and MLP_c as fine-tuning modules. The underline indicates the best result.

modifies the relationship between the diffusion timestep t and the noise intensity in the noise schedule, it is necessary to modulate the MLP_c, where its input receives the embeddings of diffusion timestep t and conditions. Conversely, adjusting the MLP layer within the transformer blocks (MLP_b) does not markedly improve the generative performance and only adds extra parameters. Hence, fine-tuning this module is not within the scope of our subsequent considerations. We also observed a decrease in model performance upon fine-tuning the **Patch Convolution (Patch Conv)** module. We hypothesize that the method of mapping 2×2 patches into a high-dimensional space remains effective for the driving dataset, as the method for mapping small patches, learned on the large-scale ImageNet dataset, demonstrates strong generalizability.

Impact of SSEI. The findings depicted in Figure 6 demonstrate that SSEI outperforms other fine-tuning methods by achieving better FID in 150 tuning iterations and enables efficiently achieving higher precision more rapidly compared to DiffFit, BitFit, and Time-Adapter. This indicates that our approach enables DiT models to converge more rapidly and attain superior generative capabilities. Note that since the VPT method is not suited for zero initialization or identity initialization, resulting in slow prompt learning, we do not plot its training curve in the graph. In Figure 6, we also explore the impact of various initialization methods on the performance of DiT model, including zero initialization, Kaiming initialization, Xavier initialization, and SSEI initialization. Our SSEI module leads to faster convergence compared to the other methods.

In Figure 7, we examine the category embeddings from ImageNet corresponding to different conditions in the Ithaca365 dataset, within the semantic encoding space of CLIP. For example, the category most closely aligned with snowy conditions is snowplow, characterized by backgrounds of expansive white roads, mirroring those found in snowy scenarios. This similarity in the semantic space suggests that embeddings with close resemblance facilitate a quicker adaptation from the

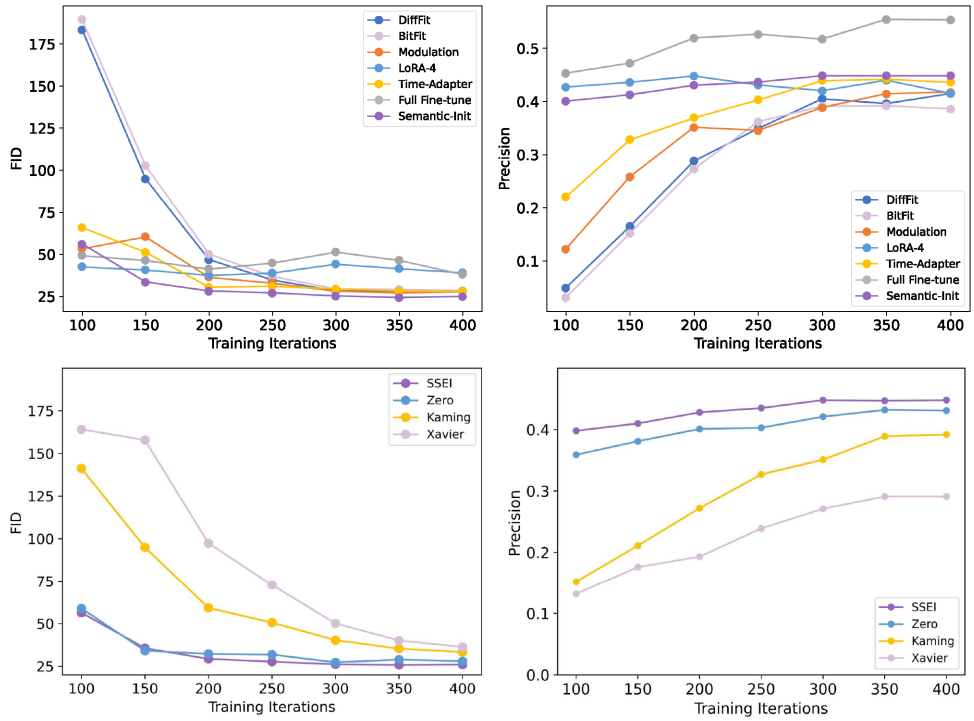

Fig. 6. The generative performance of different fine-tuning methods (row 1) and different initialization methods (row 2) in early training.

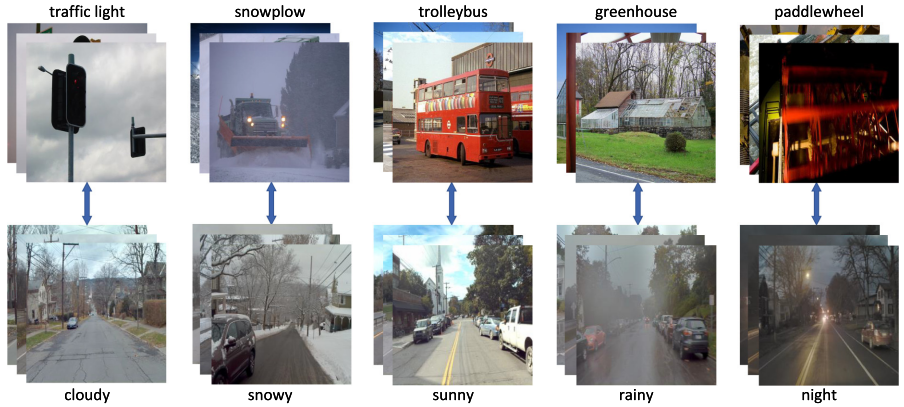

Fig. 7. Category embeddings in ImageNet that are most similar to weather and lighting condition embeddings, after using the CLIP image encoder with the cosine similarity metric.

source to the target distribution, requiring less fine-tuning time and thus enhancing the generative performance.

Impact of DriveDiTFit's Components. We conduct further ablation studies on the DriveDiTFit method, with results detailed in Table 6. Each component contributes to improve the quality and fidelity of the generated samples. The OSL increases precision from 0.448 to 0.488. This component

Table 6. Ablation Experiments on DriveDiTFit's Components

Modulation	SSEI	OSL	RoPE	PT	FID↓	sFID↓	Precision↑	Recall↑
✓					27.45	31.21	0.414	0.764
✓	✓				26.93	29.10	0.448	0.760
✓	✓	✓			25.35	29.29	0.488	0.731
✓	✓	✓	✓		23.80	29.20	0.481	0.742
✓	✓	✓	✓	✓	<u>18.64</u>	<u>26.13</u>	<u>0.626</u>	<u>0.795</u>

Each component helps improve the quality and fidelity of the generated samples.
OSL, object-sensitive loss; PT, progressive tuning; RoPE, rotary positional embeddings; SSEI, semantic-selective embedding initialization. The underline indicates the best result.

encourages the model to focus on the vehicles in driving images and improves the realism of generated images. We observe that adding RoPE in the attention mechanism reduces sFID from 25.35 to 23.80, which demonstrates its key role in maintaining the spatial relationships of local objects. It is worth noting that the implementation of the Scos noise schedule significantly boosts the generative performance by delaying the point at which vehicle details are obscured by noise. This approach provides an extended timeframe for detailed generation during the denoising process. Importantly, this progressive fine-tuning strategy is not limited to DiT models but is also applicable to the fine-tuning of other diffusion models where the original noise schedule is suboptimal.

Impact of the Backbone. In this section, we discuss the motivation of selecting DiT for autonomous driving data generation. Diffusion models include two main types of architectures—**Convolutional Neural Networks (CNN)**-based UNet architecture [18] and Transformer-based DiT architecture [31]. The CNN-based diffusion models typically employ the UNet backbone, a symmetric encoder–decoder structure. The Transformer-based diffusion models, being a more concise architecture, replace the UNet backbone with scalable standard Transformer blocks. To further validate the superiority of DiT backbone, we transferred our method to LDM [35], a CNN-based diffusion model, and compared the results with DiT models. Specifically, we fine-tune the conditional module within the UNet backbone, while keeping other modules consistent with those used in the DiT backbone.

As shown in Tables 7 and 8, the evaluation results demonstrate that our method, when applied to the Transformer-based diffusion model (DiT-XL/2), outperforms the CNN-based diffusion model (LDM) on both the Ithaca365 and BDD100K datasets. We analyze two advantages of choosing DiT for generating driving data: (a) Better capture of distance pixel dependencies in global images. Driving scene images contain both complex global (e.g., weather, lighting) and local (e.g., vehicles, traffic) information, requiring generative models to establish relationships between long- and short-range pixel dependencies. In this context, the convolution operations in CNN-based diffusion models are limited in capturing long-range dependencies. However, the attention mechanism in Transformer-based diffusion models can automatically learn these relationships during training, enabling a better understanding and generation of driving image data under complex weather and lighting conditions. (b) Better alignment with scaling law. The scaling law [10, 31] for diffusion generative models indicates that the quality of generated data is positively correlated with the computational resources and model parameters of diffusion models. CNN-based architectures often require the design of specific parameters for each convolution kernel, making the model structure less flexible and harder to scale. On the other hand, Transformer-based diffusion models, which only require stacking standard Transformer blocks, can easily increase computational resources and model parameters, enabling the generation of higher quality driving image data.

Table 7. Experimental Results of Different Backbones on the Ithaca365 Dataset

Model	Backbone	FID↓	sFID↓	Precision↑	Recall↑
LDM	UNet	65.42	42.09	0.254	0.590
DiT-XL/2	DiT	18.64	26.13	0.626	0.795

The underline indicates the best result.

Table 8. Experimental Results of Different Backbones on the BDD100K Dataset

Model	Backbone	FID↓	sFID↓	Precision↑	Recall↑
LDM	UNet	45.63	16.90	0.332	0.436
DiT-XL/2	DiT	33.38	11.93	0.659	0.611

The underline indicates the best result.

Table 9. Performance Comparison of Fine-Tuning Different Parameters on the Ithaca365 Dataset

Configuration	Params (M)	FID ↓	sFID ↓	Precision ↑	Recall ↑
Config A	0.490 (0.07%)	29.23	31.40	0.379	0.751
Config B	2.372 (0.35%)	27.45	31.21	0.414	0.764

The underline indicates the best result.

Impact of Gap-driven Fine-tuning. In this section, we conduct the experiment to validate the hypothesis that the gaps between fine-tuning datasets and original datasets influence the number of parameters that need to be adjusted. We fine-tune the DiT model on the Ithaca365 dataset using two configurations: fine-tuning the model's bias parameters (small number of parameters, Config A) and fine-tuning the model's weight parameters (large number of parameters, Config B). As shown in Table 9, the performance of Config B is superior to Config A in terms of the four metrics. This indicates that fine-tuning only a small number of bias parameters is insufficient to bridge the gap between the original dataset and the driving dataset. Therefore, our proposed DriveDiTFit utilizes Config B and fine-tune a larger number of model parameters, adapting the significant gap between the original and driving datasets.

Impact of the Fine-tuning Parameters. In Table 10, we conduct ablation experiments on various fine-tuning hyperparameters, including the rank r in modulation, fine-tuning different layers in DiT, the coefficient λ_o of the OSL, the learning rate, and the progressive variation of the noise schedule exponent s.

We conducted an analysis to determine the influence of varying the rank r in the modulation parameters on the image quality. As shown in Table 10(a), we notice a slight drop in the FID when the rank is increased from 1 to 4, where it achieves its minimum value of 18.64. This suggests that increasing the rank up to a certain point can be beneficial for the model's performance in terms of image generation quality. However, beyond rank 4, there is no substantial improvement in the FID metric. It remains relatively stable as the rank continues to increase from 4 to 10. This indicates that setting the rank to 4 is sufficient to effectively transfer the model from a pre-trained classification dataset to a driving dataset.

Table 10. Ablation Experiments of Fine-tuning Parameters on the Ithaca365 Dataset

(a) Rank r

Rank r	FID ↓
1	20.32
2	19.18
4	<u>18.64</u>
6	18.68
8	18.73
10	18.70

The underline indicates the best result.

(b) Layer l

Layers l	FID ↓
1→ 7	23.91
1→ 14	24.02
1→ 28	<u>19.55</u>
7→ 28	21.76
14→ 28	27.31
21→ 28	42.73

The underline indicates the best result.

(c) OSL Weight λ_o

λ_o	FID ↓
10	30.43
5	21.64
1	<u>19.45</u>
0.5	20.12
0.1	21.69
0.01	22.21

The underline indicates the best result.

(d) Learning Rate

LR Ratio	FID ↓
0.1×	24.21
0.5×	24.19
1×	<u>20.09</u>
5×	27.69
10×	32.66
50×	30.30

The underline indicates the best result.

(e) Noise Schedule Exponent s

Noise Schedule						FID ↓
Linear	Scos^6	Scos^5	Scos^4	Scos^3	Scos^2	
					✓	29.21
✓					✓	29.25
✓	✓				✓	24.74
✓			✓		✓	27.56
✓	✓		✓		✓	21.64
✓	✓	✓		✓	✓	24.12
✓	✓	✓	✓		✓	22.93
✓	✓	✓	✓	✓	✓	<u>19.82</u>

The underline indicates the best result.

In Table 10(b), we study the impact of fine-tuning different layers on the model's performance. Our findings reveal that fine-tuning a comprehensive range of layers, specifically from layer 1 to layer 28, achieves the lowest FID score of 19.55. This suggests that a broad spectrum of features, both shallow and deep, contributes to the model's ability to generate high-quality images for the driving dataset. Moreover, we observe that fine-tuning the initial layers (1→ 7) results in an FID score of 23.91, which is lower than the score of 42.73 when fine-tuning the deeper layers (21→ 28). Similarly, fine-tuning a slightly broader range of initial layers (1→ 14) yields an FID score of 24.02, which is better than the score of 27.31 achieved when fine-tuning the middle to deeper layers (14→ 28). These results indicate that the features extracted by the shallower layers are more critical for the generation of driving data.

In Table 10(c), we investigate the effect of scaling various OSL coefficient λ_o on the model's performance. Our findings indicate that the OSL coefficient plays a significant role in balancing the training stability and the generation of fine details in driving data. When the coefficient is set too high, it can lead to training instability, as it may overshadow the original diffusion model's loss, resulting in a degradation of the FID score and a subsequent decrease in image quality. For instance, a coefficient of 10 yields an FID score of 30.43, which is significantly higher than the optimal score, suggesting that such a high weight could be detrimental to the model's performance. Conversely, setting the coefficient too low, such as 0.01, may cause the OSL to be ineffective, failing to emphasize

the generation of vehicle details. The most optimal balance is achieved with a coefficient of 1, where the FID score is 19.45, indicating that this weight setting effectively enhances the model's ability to generate high-quality images without compromising training stability.

We conducted the influence of different learning rates on the fine-tuning process in Table 10(d). Our results indicate that the choice of learning rate significantly affects the model's ability to converge to a good performance on the quality of the generated images. Since pre-training has already initialized most of the model's parameters on million scale of datasets, most fine-tuning methods typically require a larger learning rate than pre-training [20], which helps quickly adapt the remaining parameters to the new task. Compared with pre-training learning rate of $0.1 \times (1e^{-4})$, using a learning rate of $1 \times (1e^{-3})$ yields the best FID metric. Besides, when the learning rate is increased to 5×, 10×, and 50×, the FID scores deteriorate to 27.69, 32.66, and 30.30, respectively. This trend indicates that higher learning rates lead to training instability, where the model may overshoot the optimal weights during training, causing oscillations and preventing convergence to a good solution.

In Table 10(e), we explored the effect of various exponents s of the noise schedule on the model's performance. We observed that transitioning from the linear noise schedule to the Scos^2 noise schedule through a gradual fine-tuning process yields more stable results and lower FID scores. This approach is superior to a direct shift from the linear to the Scos^2 noise schedule, which results in worse FID scores of 29.21 and 29.25, respectively. This progressive approach allows the model to adapt to the new noise characteristics in a controlled manner, which is crucial for improving the quality of image generation.

4.3 Qualitative Evaluation

To gain a comprehensive understanding of the DriveDiTFit approach, we showcase its ability to generate synthesized images across a variety of weather and lighting conditions, alongside demonstrating the effectiveness of progressive tuning strategies employing the Scos noise schedule.

As illustrated in Figure 8, the DiT model, once fine-tuned with DriveDiTFit, can generate high-quality and huge-diversity driving scenario images under given conditional embeddings. Remarkably, this model can create realistic depictions of vehicles on roads without relying on structured conditional inputs like vehicle bounding boxes [48] or segmentation maps [49].

In Figures 10 and 11, we showcase generated samples from the Ithaca365 and BDD100K datasets, employing various effective fine-tuning methods. Our proposed DriveDiTFit method is capable of generating high-quality driving data across diverse scenarios, including urban, highway, and rural environments. For BitFit, DiffFit, LoRA, and VPT, we find that the fine-tuned DiT models struggle to generate accurate vehicle details under complex weather conditions. For instance, in Figure 10, DiffFit causes distortion in the vehicle contours under snowy conditions, while VPT fails to fully generate tire details under cloudy conditions. In Figure 11, both BitFit and LoRA exhibit vehicle deformation under overcast conditions. The reason behind these facts is that these methods do not adequately account for the noise in the pre-trained model and its impact on the driving data. As a result, the time window during detail generation is insufficient, failing to ensure realism. Although the DiT model fine-tuned with Time-Adapter shows some improvement in snowy driving images, the relationship between the snow cover and the vehicle remains unrealistic. Similarly, Full Fine-tune also suffers from vehicle deformation issues, especially under complex environmental conditions. In contrast, the DiT model fine-tuned with DriveDiTFit not only learns significant weather and lighting characteristics but also captures scene details, including car headlight halos in the night scenario of Figure 10, complete vehicle outlines in each scenario of Figure 11, and the physical relationships between rain, snow, and the vehicle in the snowy and

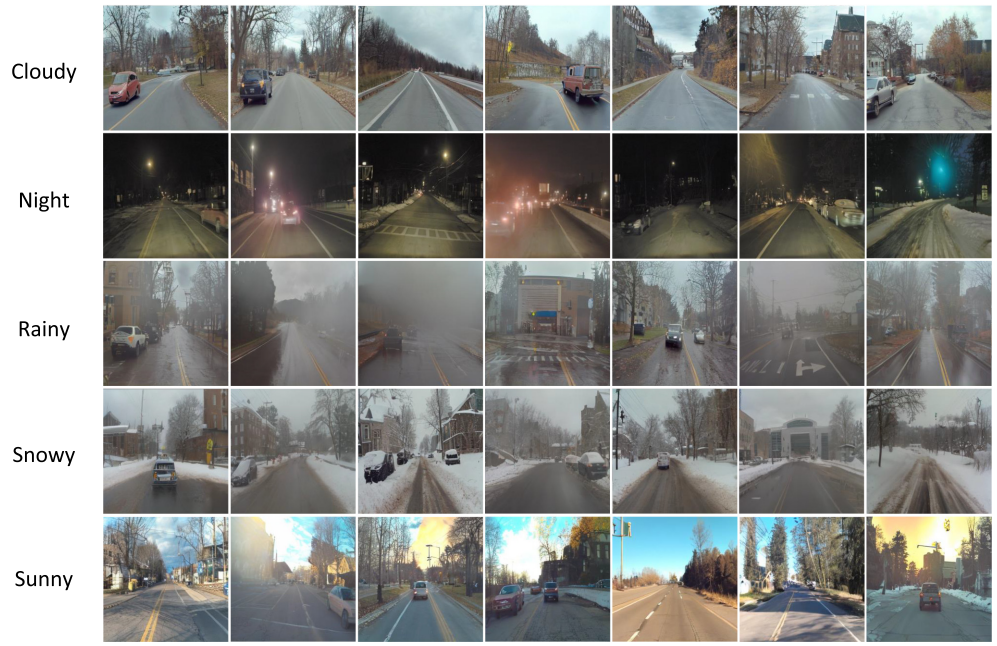

Fig. 8. Samples from DiTs fine-tuned on the Ithaca365 datasets through the DriveDiTFit method, each with a resolution of 256 × 256.

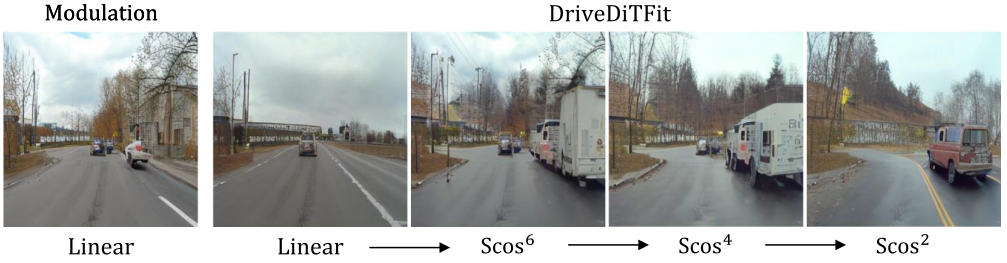

Fig. 9. The process visualization of progressive tuning strategies with the Scos noise schedule.

rainy scenario of Figure 10. These observations validate the reliability of our proposed fine-tuning method.

We further present the impact of the DriveDiTFit components, with a particular focus on the generation of details at various phases of the progressive tuning process. The visualization in Figure 9 serves to validate the suitability of the Scos noise schedule for driving datasets. Employing a consistent noise, the integration of vehicle masks and the RoPE module, which leverages local attention mechanisms, significantly contributes to the accurate formation of complete vehicle contours, in contrast to traditional modulation methods. Furthermore, the use of the Scos noise schedule in the progressive tuning regimen notably refines the representation of intricate details, methodically enhancing the visualization of complex features such as tires, lane markings, and windows, thereby demonstrating the effectiveness of this approach in producing highly detailed and realistic images.

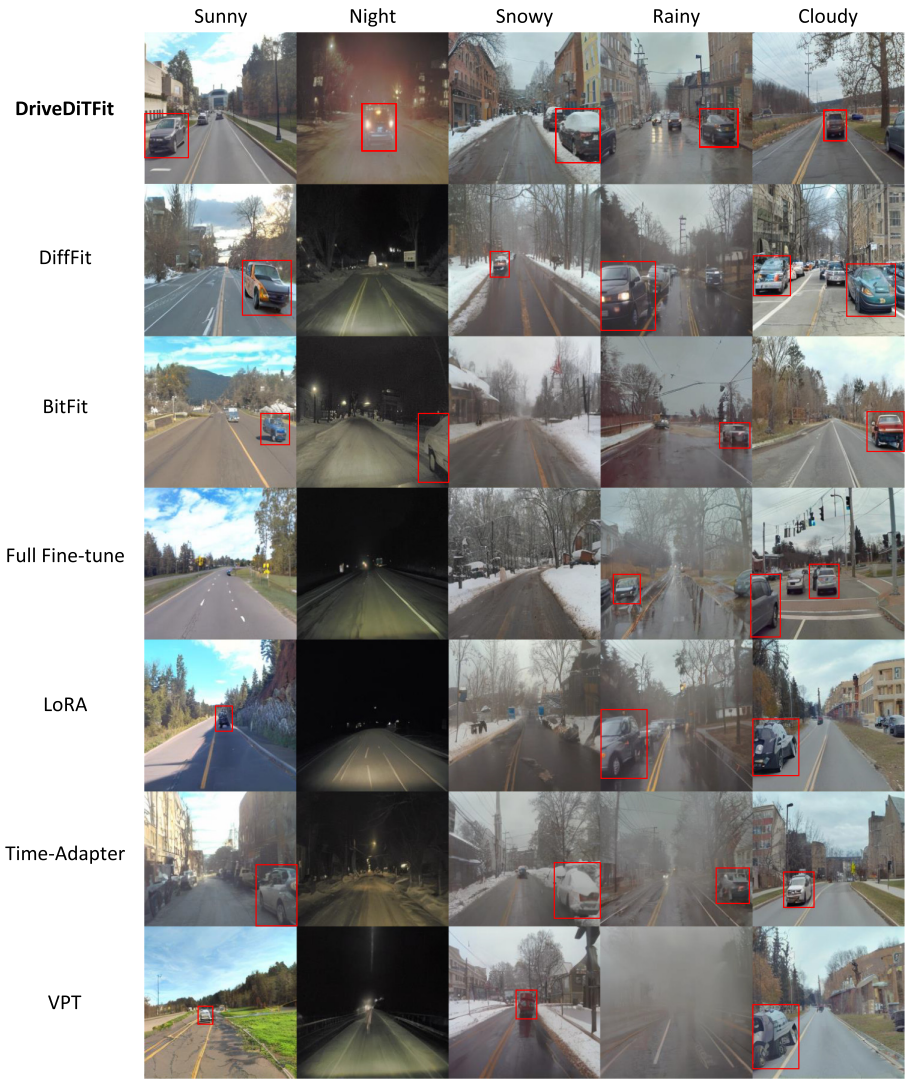

Fig. 10. Comparing details of generated samples between different methods on the Ithaca365 dataset.

4.4 Extension on Classification Datasets

In this section, we conduct experiments on classification datasets to validate the generality of our method. Following the work of Xie et al. [51], we choose two commonly used classification datasets: Caltech [15] and Oxford Flowers [30]. The Caltech dataset comprises photos of objects within 101 distinct categories, while the Oxford Flowers dataset consists of 102 categories of fine-grained classification images of flowers. We report FID metric using 50 sampling steps and set the classifier guidance to 1.5. As shown in Table 11, our method outperformed other traditional fine-tuning method by a margin across four metrics. This indicates that our method is not limited to driving scenarios and can also be extended to other types of datasets, including both coarse-grained and fine-grained classification datasets.

Fig. 11. Comparing details of generated samples between different methods on the BDD100K dataset.

4.5 Downstream Task Evaluation

To validate that the generated dataset benefits downstream perception tasks in autonomous driving applications, particularly in enhancing the performance of visual perception models for imbalanced categories (such as adverse weather and lighting conditions), we choose the object detection task using the YOLOv8 [43] model. We train the model separately on a real dataset and a mixed dataset (real + generated data) using default training parameters on a single 4090 GPU. The trained object detection models are then evaluated on a validation set consisting of adverse weather and lighting conditions, including rainy, night, and snowy environments. As shown in Table 12, compared to the real dataset, the generated dataset improved the YOLOv8 model's performance by 0.021 on the Ithaca365 dataset, by 0.019 on the BDD100K dataset, and by 0.031 on the nuScenes dataset. This demonstrates that the proposed method's generated dataset can enhance the YOLOv8 model's performance in object detection tasks, further confirming that the generated dataset is beneficial for downstream tasks.

Table 11. The Performance Comparison of Different Fine-tuning Methods on the Caltech and Oxford Flowers Dataset

Method	Caltech	Oxford Flowers
Full Fine-tune	35.25	21.05
Time-Adapter	35.83	20.98
LoRA-R8	86.05	164.13
VPT-Deep	42.78	25.59
BitFit	34.21	20.31
DiffFit	33.84	20.18
Ours	<u>28.85</u>	<u>17.35</u>

All FID results, except those for our method and Time-Adapter, are sourced from [51]. The underline indicates the best result.

Table 12. Validation Results for the Effectiveness of the Generated Dataset in the Object Detection Task

Datasets	Ithaca365		BDD100K		nuScenes	
Real	Mixed	Real	Mixed	Real	Mixed	
mAP@50	0.502	<u>0.523</u>	0.395	<u>0.414</u>	0.367	<u>0.398</u>

The underline indicates the best result.

5 Conclusion

In this article, we introduce DriveDiTFit, an innovative approach for efficient generation of autonomous driving data through fine-tuning pre-trained DiTs. From the model architecture perspective, we have developed a modulation technique designed to adjust the overall style and layout of the generated samples, making it suitable for autonomous driving datasets that significantly diverge from pre-trained dataset. Building on this, we have implemented an SSEI approach to help the model more rapidly learn the data distribution and achieve superior generative performance. Notably, we are pioneers in identifying the influence of the original noise schedule on downstream datasets and have introduced a novel Scos noise schedule and an OSL coupled with a progressive fine-tuning strategy to enhance the detail generation of small objects in scenarios. Our experiments demonstrate that our proposed method can produce high-quality, diverse driving datasets under various weather and lighting conditions.

References

[1] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindström, Daria Motorniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. 2023. Zenseact Open Dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 20178–20188.

[2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22669–22679.

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22563–22575.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11621–11631.

[5] Peishan Cong, Yiteng Xu, Yiming Ren, Juze Zhang, Lan Xu, Jingya Wang, Jingyi Yu, and Yuexin Ma. 2023. Weakly supervised 3d multi-person pose estimation for large-scale scenes based on monocular camera and single LiDAR. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 461–469.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. Retrieved from https://arxiv.org/abs/1810.04805

[8] Carlos A. Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, et al. 2022. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21383–21392.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929. Retrieved from https://arxiv.org/abs/2010.11929

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*.

[11] Qian Feng, Hanbin Zhao, Chao Zhang, Jiahua Dong, Henghui Ding, Yu-Gang Jiang, and Hui Qian. 2024. PECTP: Parameter-efficient cross-task prompts for incremental vision transformer. arXiv:2407.03813. Retrieved from https://arxiv.org/abs/2407.03813

[12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv:2208.01618. Retrieved from https://arxiv.org/abs/2208.01618

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3354–3361.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM* 63, 11 (2020), 139–144.

[15] Gregory Griffin, Alex Holub, and Pietro Perona. 2007. *Caltech-256 Object Category Dataset*. Technical Report 7694. California Institute of Technology, Pasadena, CA.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems* 30 (2017), 6629–6640.

[17] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, et al. 2022. Imagen video: High definition video generation with diffusion models. arXiv:2210.02303. Retrieved from https://arxiv.org/abs/2210.02303

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

[19] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. 2023. Simple diffusion: End-to-end diffusion for high resolution images. arXiv:2301.11093. Retrieved from https://arxiv.org/abs/2301.11093

[20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. arXiv:2106.09685. Retrieved from https://arxiv.org/abs/2106.09685

[21] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 1501–1510.

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision*. Springer, 709–727.

[23] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv:1312.6114. Retrieved from https://arxiv.org/abs/1312.6114

[24] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning Multiple Layers of Features from Tiny Images*. Master's thesis, University of Tront.

[25] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems* 32 (2019), 3927–3936.

[26] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv:2108.01073. Retrieved from https://arxiv.org/abs/2108.01073

[27] Taehong Moon, Moonseok Choi, Gayoung Lee, Jung-Woo Ha, and Juho Lee. 2022. Fine-tuning diffusion models with limited data. In *Proceedings of the NeurIPS 2022 Workshop on Score-Based Methods*.

[28] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. 2021. Generating images with sparse representations. arXiv:2103.03841. Retrieved from https://arxiv.org/abs/2103.03841

[29] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 8162–8171.

[30] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Proceedings of the 2008 6th Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 722–729.

[31] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*. PMLR, 8748–8763.

[33] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. arXiv:2204.06125. Retrieved from https://arxiv.org/abs/2204.06125

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI '15), part III*. Springer, 234–241.

[37] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22500–22510.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. arXiv:2010.02502. Retrieved from https://arxiv.org/abs/2010.02502

[39] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063.

[40] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.

[41] Runzhou Tao, Wencheng Han, Zhongying Qiu, Cheng-zhong Xu, and Jianbing Shen. 2023. Weakly supervised monocular 3D object detection using multi-view projection and direction consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17482–17492.

[42] Jiahang Tu, Hao Fu, Fengyu Yang, Hanbin Zhao, Chao Zhang, and Hui Qian. 2024. Texttoucher: Fine-grained text-to-touch generation. arXiv:2409.05427. Retrieved from https://arxiv.org/abs/2409.05427

[43] Rejin Varghese and M. Sambath 2024. YOLOv8: A novel object detection algorithm with enhanced performance and robustness. In *Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 1–6. DOI: https://doi.org/10.1109/ADICS58448.2024.10533619

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017), 6000–6010.

[45] Boyang Wang, Bowen Liu, Shiyu Liu, and Fengyu Yang. 2024. VCISR: Blind single image super-resolution with video compression synthetic data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4302–4312.

[46] Boyang Wang, Fengyu Yang, Xihang Yu, Chao Zhang, and Hanbin Zhao. 2024. APISR: Anime production inspired real-world anime super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25574–25584.

[47] Fangyikang Wang, Huminhao Zhu, Chao Zhang, Hanbin Zhao, and Hui Qian. 2024. GAD-PVI: A general accelerated dynamic-weight particle-based variational inference framework. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 15466–15473.

[48] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. 2023. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. arXiv:2311.17918. Retrieved from https://arxiv.org/abs/2311.17918

[49] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. 2023. Panacea: Panoramic and controllable video generation for autonomous driving. arXiv:2311.16813. Retrieved from https://arxiv.org/abs/2311.16813

[50] Xintian Wu, Hanbin Zhao, Liangli Zheng, Shouhong Ding, and Xi Li. 2022. Adma-GAN: Attribute-Driven Memory Augmented GANs for text-to-image generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1593–1602.

[51] Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. 2023. DiffFit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4230–4239.

[52] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[53] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv:2106.10199. Retrieved from https://arxiv.org/abs/2106.10199

[54] Huminhao Zhu, Fangyikang Wang, Chao Zhang, Hanbin Zhao, and Hui Qian. 2024. Neural Sinkhorn gradient flow. arXiv:2401.14069. Retrieved from https://arxiv.org/abs/2401.14069

Appendix

A Additional Generated Samples

We present more generated samples on the Ithaca365 and BDD100K datasets, respectively, in Figures A1 and A2, using different fine-tuning methods, which thoroughly validates our proposed DriveDiTFit method's capability to generate high-quality driving data under diverse weather and lighting conditions.

(a) DriveDiTFit

(b) DiffFit

(c) BitFit

Fig. A1. Additional generated samples on the Ithaca365 dataset—part 1, part 2, and part 3.

(d) LoRA-4

(e) Time-Adapter

(f) Full Fine-tune

Fig. A1. Continued

(g) VPT

Fig. A1. Continued

(a) DriveDiTFit

(b) DiffFit

Fig. A2. Additional generated samples on the BDD100K dataset—part 1, part 2, and part 3.

(c) BitFit

(d) LoRA-4

(e) Time-Adapter

Fig. A2. Continued

(f) Full Fine-tune

(g) VPT

Fig. A2. Continued

Received 23 July 2024; revised 29 December 2024; accepted 5 January 2025