arXiv:2507.00236v1 [cs.RO] 30 Jun 2025

Sim2Real Diffusion: Learning Cross-Domain Adaptive Representations for Transferable Autonomous Driving

Chinmay Samak[⋆†], Tanmay Samak[⋆†], Bing Li[†], and Venkat Krovi[†]

Abstract*— Simulation-based design, optimization, and validation of autonomous driving algorithms have proven to be crucial for their iterative improvement over the years. Nevertheless, the ultimate measure of effectiveness is their successful transition from simulation to reality (sim2real). However, existing sim2real transfer methods struggle to comprehensively address the autonomy-oriented requirements of balancing: (i) conditioned domain adaptation, (ii) robust performance with limited examples, (iii) modularity in handling multiple domain representations, and (iv) real-time performance. To alleviate these pain points, we present a unified framework for learning cross-domain adaptive representations for sim2real transferable autonomous driving algorithms using conditional latent diffusion models. Our framework offers options to leverage: (i) alternate foundation models, (ii) a few-shot fine-tuning pipeline, and (iii) textual as well as image prompts for mapping across given source and target domains. It is also capable of generating diverse high-quality samples when diffusing across parameter spaces such as times of day, weather conditions, seasons, and operational design domains. We systematically analyze the presented framework and report our findings in the form of critical quantitative metrics and ablation studies, as well as insightful qualitative examples and remarks. Additionally, we demonstrate the serviceability of the proposed approach in bridging the sim2real gap for end-to-end autonomous driving using a behavioral cloning case study. Our experiments indicate that the proposed framework is capable of bridging the perceptual sim2real gap by over 40%. We hope that our approach underscores the potential of generative diffusion models in sim2real transfer, offering a pathway toward more robust and adaptive autonomous driving.

I. Introduction

Development of autonomous driving systems demands rigorous training, fine-tuning, and thorough testing across diverse scenarios to guarantee safety, reliability, and scalability. However, conducting these processes in real-world environments is often constrained by significant expenses, time commitments, safety risks, and limited ability to replicate rare or extreme conditions. In such a milieu, simulation frameworks present a compelling case by offering a cost-effective space for training and validation while alleviating monetary, safety, spatial, and temporal constraints imposed during physical testing [1]. Simulations provide control over test case generation and execution, enabling testing in controlled settings that effectively account for variability [2]–[4]. These controlled settings are extremely crucial for comprehensive corner-case analysis, especially during safety-critical testing involving social and situational variability [5]–[7]. Another important autonomy-oriented utility of simulations is generating synthetic data, which facilitates rich training and robust testing across a range of variations [8]–[12]. Furthermore, simulations enable reproducibility and benchmarking, ensuring that results are consistent and repeatable across the same experiments conducted at different points in time [13]–[15]. Finally, the ability of simulations to parallelize training and testing workloads accelerates the overall development-validation process, while allowing for flexible and scalable solutions that can be deployed in the cloud [2], [3], [16]–[18].

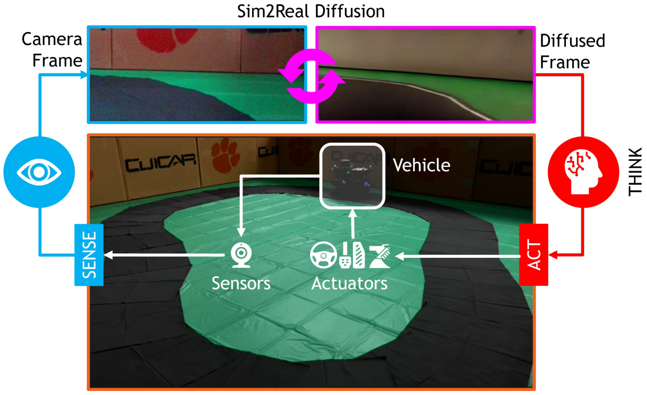


Fig. 1: Proposed sim2real diffusion approach for autonomous driving depicting the vehicle performing end-to-end navigation using recursive perception, domain adaptation, planning, and control processes in real-time.

However, despite all the benefits simulation has to offer, it is important to note that even perfectly functioning autonomous vehicles in the simulation (across all parameter sweeps, edge-cases, etc.) pose no practical benefit to society unless they can work with similar reliability in the real world, and oftentimes, performance in simulation does not necessarily translate to success in the real world due to the inherent simulation-to-reality (sim2real) gap. Autonomy algorithms that operate flawlessly in simulation frequently suffer from performance degradation when exposed to real-world variability and uncertainty. This discrepancy results either at the perception interface (e.g., camera images not photorealistic enough), or the control interface (e.g., vehicle dynamics not accurate enough), or across both the interfaces. In this work, we focus on the perceptual domain gap between simulation and reality (sim2real gap).

[⋆] These authors contributed equally.

[†] Department of Automotive Engineering, Clemson University International Center for Automotive Research (CU-ICAR), Greenville, SC 29607, USA. {csamak, tsamak, bli4, vkrovi}@clemson.edu

Previous related works addressing this problem can be broadly categorized based on their isolated focus on identification, adaptation, and augmentation of domains (simulation or reality), as we will review below.

A. Domain Identification

The main idea behind domain identification is to identify the critical parameters of the target domain (real world) and calibrate the source domain (simulation) to match those parameters, so as to closely capture the target domain data distribution within the source domain. A range of domain identification techniques have been explored in prior research, each differing in its level of complexity, precision, and practical application.

Manual handcrafting (legacy method), while capable of delivering exceptional results, is notably time-intensive and demands considerable human effort and proficiency. Moreover, this approach can introduce human biases and errors that negatively affect the domain representations.

Surface reconstruction techniques such as those described in [19], [20] produce seamless and continuous 3D surfaces from sparse point clouds. These methods are typically precise but can falter when processing noisy or incomplete datasets and generally lack the ability to capture photorealistic textures or appearances. Photogrammetry approaches [21], [22] build 3D structures by analyzing multiple images taken from different viewpoints, leveraging their overlaps and spatial relationships. These techniques can closely replicate both geometric form and visual detail but require accurate image acquisition and may be less effective when applied to large-scale scenes or objects.

In contrast, neural radiance fields (NeRFs) [23], [24] utilize deep learning to model volumetric scenes from a limited number of views, generating highly photorealistic 3D outputs. Despite their strengths in capturing intricate details and textures, NeRFs are computationally demanding and significantly slower compared to alternative methods. 3D Gaussian splatting (3DGS) [25]–[28] addresses these limitations by offering a faster and more memory-efficient strategy. It represents 3D points using Gaussian splats defined by parameters such as position, scale, color, and opacity, collectively shaping the visual and geometric outcome. This technique has shown superior performance over traditional voxel-based methods, especially in rendering expansive environments, making it highly suitable for applications in autonomous driving simulations.

To summarize, the main benefits of these domain identification methods are that they offer the flexibility of adopting physics-based and/or data-driven approaches for modeling, and the derived representations are temporally and/or semantically consistent. However, this usually comes at the price of requiring hand-tuning a lot of parameters (typically from domain knowledge) that govern the process and requirement of vast amounts of high-quality datasets capturing the target domain, not to mention there is always a fidelity vs. real-time performance tradeoff.

B. Domain Adaptation

The key concept of domain adaptation is to adapt algorithms trained (or tuned/optimized) in the source domain to perform well on a different but related target domain, where the data distributions differ. This is achieved through the learning of the statistical differences between the source and target domains, and numerous approaches to domain adaptation have been investigated in earlier studies.

Transfer learning techniques [29]–[31] are one of the most commonly used methods for domain adaptation, where models trained on large datasets from the source domain are re-trained or fine-tuned on a few examples from the target domain, typically by freezing some of the model parameters. This enables learning major differences in feature representations without necessarily overfitting the data.

Another commonly used approach for achieving domain adaptation is curriculum learning [32]–[35], where the idea is to start learning in the source domain and then sequentially complicate the learning objective (in the direction of the target domain) so that the model can gradually learn to adapt to the statistical differences in data distributions, rather than attempting to do this in a single shot.

Meta-learning [36]–[39], often referred to as *"learning to learn"*, is used in domain adaptation to enable models to quickly adapt to new domains with limited data. In this context, meta-learning trains a model across a variety of tasks or domains so that it can learn a generalizable adaptation strategy. This training process equips the model with the ability to rapidly fine-tune itself to a new, unseen target domain using only a small number of labeled examples. Techniques like model-agnostic meta-learning [40]–[43] are commonly used, where the model learns parameters that are sensitive to changes in domain, allowing fast adaptation with minimal updates. By focusing on learning adaptable patterns rather than domain-specific ones, meta-learning helps overcome the domain shift problem and improves generalization in scenarios where labeled target data is scarce.

Finally, knowledge distillation [44]–[47] offers a more indirect approach to domain adaptation by transferring knowledge from a well-trained teacher model (usually trained on the source domain) to a student model targeting the new domain. In this setup, the teacher provides "soft labels" or output probabilities that contain richer information than hard labels alone, helping the student model learn better generalization despite domain gaps. This process allows the student to mimic the teacher's behavior while adapting to the specific characteristics of the target domain, even with limited labeled data. Knowledge distillation is especially effective when operating under size, weight, and power (SWaP) constraints, such as computational limitations.

In summary, domain adaptation offers benefits such as high-performance training in simulation (source domain) and data-driven adaptation to the real world (target domain). However, this naturally makes these approaches data-dependent, requiring additional effort in order to achieve sim2real transfer.

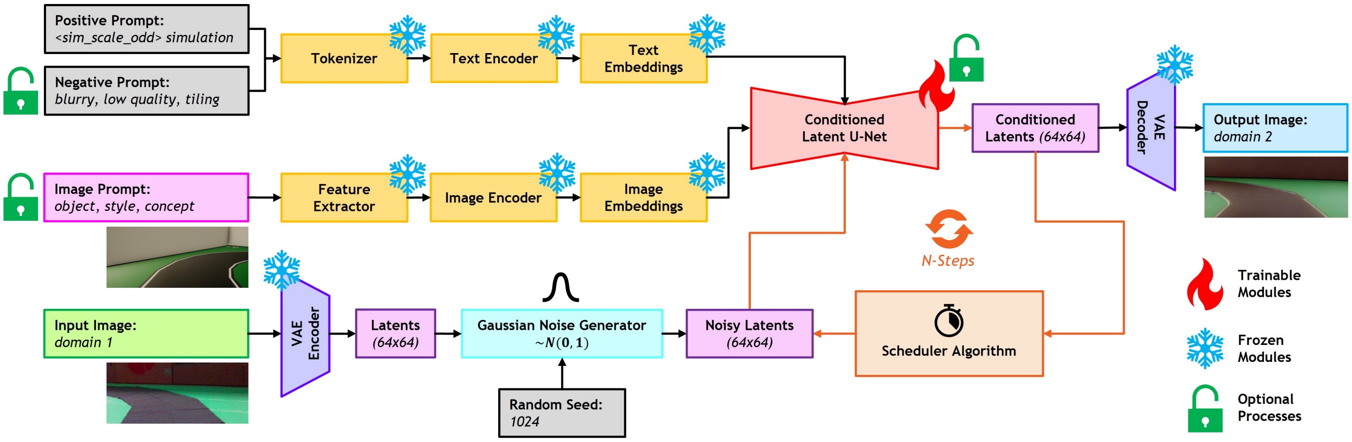


Fig. 2: Proposed framework for enabling sim2real transfer of autonomous driving algorithms through the learning of adaptive cross-domain representations using a combination of image and text conditioning within a latent diffusion architecture.

C. Domain Augmentation

Domain augmentation seeks to address the domain gap issue by learning under an inflated source domain, which is expected to capture the target domain distribution. A variety of methods exist herein, which have been explored in earlier research as discussed below.

Robust learning [48]–[50] focuses on improving model generalizability across a variety of augmented data, especially when the domain is subject to perturbations or noise. By employing techniques like adversarial perturbations, noise injection, or robust loss functions, the model learns to focus on the core features that are most relevant to the task, rather than overfitting to domain-specific characteristics.

A similar method, domain randomization [51]–[53], involves explicitly generating a wide variety of synthetic training data by randomly varying domain-specific attributes such as texture, lighting, background, and camera angles. This randomness forces the model to learn from a diverse set of scenarios, thereby improving its generalizability.

Finally, style transfer techniques [54]–[58] manipulate the appearance of one domain by applying the style of a different domain. This enables the model to learn the domain-invariant features by exposing it to data that maintains structural integrity while adopting a different style, such as changes in texture, color, or artistic features. While robust learning enhances generalization through noise reduction and domain randomization increases diversity in training data, style transfer emphasizes learning useful features across different visual representations, thus boosting domain adaptation through controlled visual transformations.

In all, augmentation methods usually enjoy the benefit of requiring additional efforts only within the source domain (simulation). However, it is often impossible to accurately or even realistically capture the target domain (real world) conditions in simulation with probabilistic augmentations. This usually results in a low training performance (due to added simulation variability), and the trained algorithms, being data-dependent, typically cannot be guaranteed to work well in the real world.

D. Novel Contributions

We propose a unified framework for enabling sim2real transfer of autonomous driving algorithms (refer Fig. 1 and 2) by learning adaptive cross-domain representations using conditional latent diffusion models. Our framework is designed to meet key autonomy-oriented sim2real transfer requirements: (i) accurate cross-domain mapping to preserve semantic and geometric consistency between domains, (ii) rapid adaptation to novel domains using few-shot learning, (iii) robust and modular handling of diverse sets of source and target domains, and (iv) real-time operation for compatibility with closed-loop autonomy algorithms.

The core novelty of the framework lies in its ability to diffuse high-fidelity representations of driving scenarios across domains, utilizing a mix of image and text conditioning within a latent diffusion architecture. By incorporating both foundational vision-language models and domain-specific fine-tuning pipelines, the framework supports efficient mapping between manifolds of simulated and real-world images – even across variations such as times of day, weathers, seasons, and operational design domains. Unlike existing sim2real transfer methods, our method offers a flexible and modular sim2real diffusion adapter, which decouples the core autonomy algorithm(s) from cross-domain adaptation. The proposed framework thus enables efficient sim2real transferability while significantly reducing dependency on large datasets (for retraining) or multiple models (for different domain representations).

The key contributions of this paper are summarized below:

- **Sim2Real Diffusion:** We provide a conditional latent diffusion pipeline capable of mapping simulation features onto real-world camera frames at runtime. The said pipeline preserves semantic and geometric relations across domains via controlled conditioning, and allows few-shot training of novel domain representations for robust and modular adaptation. Finally, the proposed pipeline supports real-time operation for online inference with autonomous driving systems.

- **Performance Evaluation:** We explore the conditioned generation and domain adaptation capabilities of the latent diffusion model. Additionally, we also perform ablation studies and computational analysis to qualitatively and quantitatively analyze the effect of various components of the pipeline on the domain adaptation.
- **Case Study:** We present an end-to-end autonomous driving case study of behavioral cloning trained using simulation-only data (without any data augmentations). We employ the proposed framework to project the real-time camera feed from the real-world domain back onto the simulation manifold, thereby achieving sim2real diffusion. The behavioral cloning model inference is run on the diffused frames to bridge the sim2real gap.

II. Research Methodology

A. Sim2Real Diffusion

The proposed sim2real diffusion framework (refer Fig. 2) is a conditional latent diffusion model (LDM) [59], which addresses the two key challenges of diffusion probabilistic models (DPM) [60], viz., low inference speed and very high training costs. The key difference between the two model architectures is that DPMs operate directly in the high-dimensional pixel space, while LDMs operate in a latent space derived through perceptual image compression using an autoencoder optimized with a fusion of a perceptual loss and a patch-based adversarial objective. This latent space is better suited for likelihood-based generative models because it allows them to (i) concentrate on essential semantic information and (ii) train more efficiently due to the reduced dimensionality. Particularly, the input image $x \in \mathbb{R}^{H \times W \times 3}$, represented in RGB format, is transformed by the encoder $\mathcal{E}$ into a latent vector $z = \mathcal{E}(x)$. This latent representation $z \in \mathbb{R}^{h \times w \times c}$ is later passed through the decoder $\mathcal{D}$, which reconstructs the image $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$.

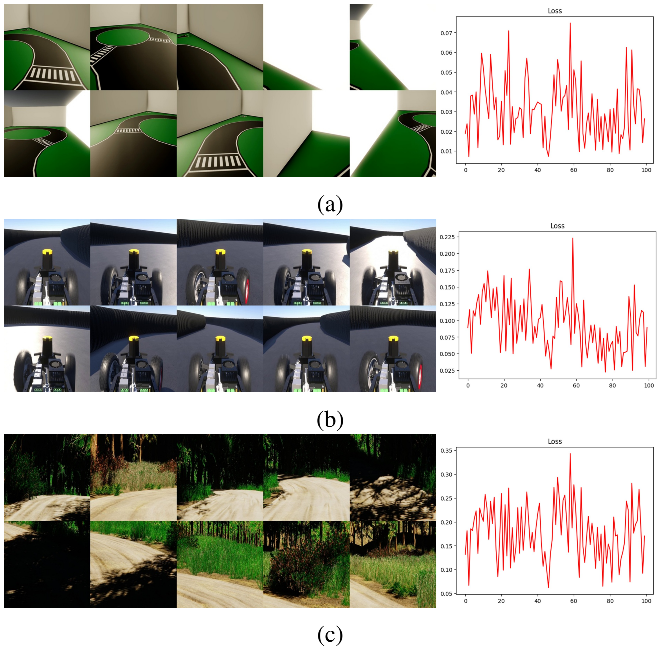
Diffusion models are probabilistic frameworks that aim to learn a denoising process by reversing the diffusion process, a Markov chain of length T that incrementally adds noise to the data. This effectively trains the model to learn the original data distribution $p(x)$ by denoising Gaussian noise over multiple steps. These models can be viewed as a sequence of denoising autoencoders $\epsilon_\theta(x_t, t)$, each trained to recover cleaner inputs from progressively noisier ones. The training objective, which is a reweighted version of the variational lower bound on $p(x)$, is simplified by varying the noise step t uniformly, i.e., $t = \{1, ..., T\}$:

$$\mathcal{L}_{\text{DPM}} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right] \tag{1}$$

In the context of LDMs, Eq. 1 can be rewritten to include the latent representation:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right] \tag{2}$$

The core neural architecture $\epsilon_\theta(\circ, t)$ of the diffusion model is a time-conditioned U-Net [61]. The U-Net comprises ResNet-based encoder and decoder blocks. The encoder downsamples the latent image, while the decoder reconstructs a cleaner, up-sampled (latent dimensions) version by predicting the noise residual. The output of the U-Net, being the noise residual, is used to compute a denoised latent image representation via a scheduler algorithm (e.g., DPM-Solver [62]). The corresponding layers of the encoder and decoder are linked via skip connections to preserve critical features during downsampling. Given that the forward diffusion process (i.e., noising) is predetermined, the latent variable z_t can be efficiently derived from the encoder $\mathcal{E}$ during training. Similarly, a single pass through the decoder $\mathcal{D}$ is sufficient to transform samples from the latent space $p(z)$ back into the image space.



(a)

(b)

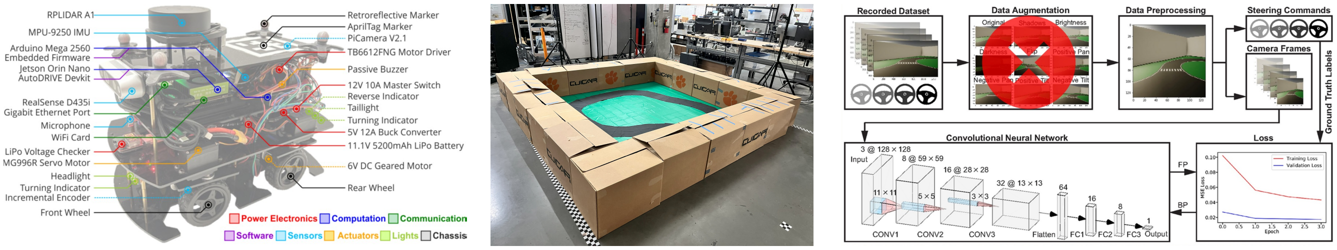
(c)

Fig. 3: Few-shot examples and fine-tuning loss for (a) Nigel (small-scale, on-road); (b) RoboRacer (small-scale, racing); (c) RZR (large-scale, off-road).

Controlling the generative process of latent diffusion models through auxiliary inputs y such as text prompts, images, semantic maps, depth maps, etc., is possible, but requires modeling conditional probability distributions of the form $p(z|y)$. This can be realized with a conditional denoising autoencoder $\epsilon_\theta(z_t, t, y)$, which in the context of U-Net, can be implemented via the cross-attention mechanism [63]. Particularly, a domain-specific encoder τ_θ is used to project multi-modal condition y onto an intermediate manifold $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$, which is then injected between the ResNet blocks of the U-Net via the cross-attention mechanism:

$$\texttt{Attention}(Q, K, V) = \texttt{Softmax}\left(\frac{Q \cdot K^\top}{\sqrt{d}}\right) \cdot V \tag{3}$$

where, $Q = W_Q^{(i)} \cdot \varphi_i(z_t)$, $K = W_K^{(i)} \cdot \tau_\theta(y)$, $V = W_V^{(i)} \cdot \tau_\theta(y)$ such that $W_Q^{(i)} \in \mathbb{R}^{d \times d_\tau}$, $W_K^{(i)} \in \mathbb{R}^{d \times d_\tau}$, $W_V^{(i)} \in \mathbb{R}^{d \times d_\epsilon^i}$ are learnable projection matrices, and $\varphi_i(z_t) \in \mathbb{R}^{N \times d_\epsilon^i}$ denotes a flattened intermediate representation of U-Net ϵ_θ.



(a)

(b)

(c)

Fig. 4: Experimental setup for the end-to-end autonomous driving case study: (a) Nigel, the small-scale vehicle with its components and sub-systems annotated; (b) environment constructed in lab; (c) training pipeline for the autonomy algorithm.

In the context of conditional LDMs, Eq. 2 can be rewritten to include the condition y by jointly optimizing τ_θ and ϵ_θ:

$$\mathcal{L}_{\text{C-LDM}} = \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right] \quad (4)$$

Additionally, the proposed pipeline also supports IP-Adapter [64], which employs a decoupled cross-attention mechanism for separating the text y_{txt} and image y_{img} features. Consequently, we have an additional term in Eq. 3, corresponding to the image prompt:

$$\texttt{Softmax}\left(\frac{Q \cdot K^\top}{\sqrt{d}}\right) \cdot V + \texttt{Softmax}\left(\frac{Q \cdot (K')^\top}{\sqrt{d}}\right) \cdot V' \quad (5)$$

where, $Q = W_Q^{(i)} \cdot \varphi_i(z_t)$, $K = W_K^{(i)} \cdot \tau_\theta(y_{\text{txt}})$, $V = W_V^{(i)} \cdot \tau_\theta(y_{\text{txt}})$, $K' = W_K'^{(i)} \cdot \tau_\theta(y_{\text{img}})$, and $V' = W_V'^{(i)} \cdot \tau_\theta(y_{\text{img}})$.

In this context, Eq. 4 can be rewritten to include the IP-Adapter:

$$\mathcal{L}_{\text{IPC-LDM}} = \mathbb{E}_{\mathcal{E}(x),y_{\text{txt}},y_{\text{img}},\epsilon\sim\mathcal{N}(0,1),t} \left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y_{\text{txt}}), \tau_\theta(y_{\text{img}}))\|_2^2\right] \quad (6)$$

It is worth mentioning that the conditional prompts (both text and image) are encoded into an embedding space that can be understood by the U-Net. Text prompts are encoded by a simple transformer-based encoder (e.g., CLIPTextModel [65]) that maps a sequence of input tokens to a sequence of latent text embeddings. Similarly, image prompts are encoded into patch image embeddings by a vision transformer-based encoder (e.g., OpenCLIP-ViT-H-14 [66]).

B. *Design of Experiments*

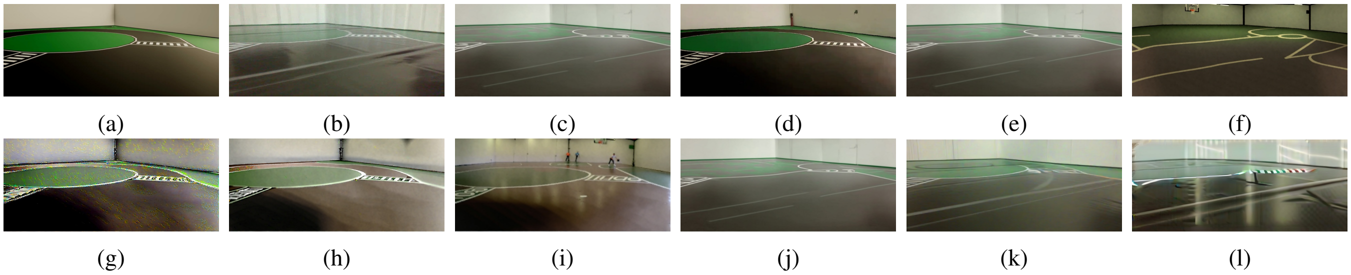
All the experiments in this work were conducted with the help of AutoDRIVE Ecosystem[1] [67], [68]. Particularly, AutoDRIVE Simulator [69], [70] was employed to simulate the autonomy-oriented digital twins [71] of Nigel (small-scale, on-road), RoboRacer (small-scale, racing), and RZR (large-scale, off-road) to collect training and testing data for the various ablation studies. Additionally, AutoDRIVE Testbed (with Nigel [72]) was used for the end-to-end autonomous driving case study (refer Fig. 4).

1) Ablation Studies: We performed ablation studies in 4 stages. The first set of experiments was meant to qualitatively assess the effect of (i) domain adaptation direction {sim2real, real2sim}, (ii) foundation model {SDXL, SDXL-Turbo}, (iii) denoising steps {1, 5, 30}, and (iv) input resolution {640×274, 1280×548, 2560×1096}. The second, third, and fourth sets of experiments were meant to assess the effect of (i) fine-tuning and (ii) image prompting, qualitatively and quantitatively. To this end, we fine-tuned the SD1.5 foundation model on 3 custom concepts (refer Fig. 3) using the DreamBooth approach [73]. The fine-tuning comprised a *"trigger word"* of the form `<sim_scale_odd>` to identify/associate a particular concept with a particular simulator, scale, and operational design domain (ODD).

2) Performance Study: We analyzed the performance of the fine-tuned latent diffusion model conditioned via text as well as image prompts. This experiment involved training and inference of the sim2real diffusion model on (i) Palmetto Cluster (8 CPU cores, 64 GB RAM) with 4 different GPU models {V100, A100, H100, L40S}, (ii) Google Colab (2 CPU cores, 12 GB RAM) with T4 GPU, and (iii) a laptop PC (20 CPU cores, 32 GB RAM) with 3080 Ti GPU.

3) Case Study: The chosen case study builds on [74], aiming to clone the end-to-end driving behavior of a human driver using a six-layer convolutional neural network (CNN). AutoDRIVE Simulator was employed for collecting 5 laps worth of manual driving data (∼1700 samples), which was balanced, normalized, and resized without any augmentation. The CNN model was trained to predict the actuator commands corresponding to the given camera frame for 4 epochs at a learning rate of 1e-3, which resulted in stable convergence. The simulation environment consisted of a figure-8 track with a black road on green vinyl mat surrounded by white boxes. The road had white lane markings, and the intersection had marked crosswalks. Contrarily, the real-world environment consisted of a rather irregular track created using black tissues (textures, flat edges, folds) laid out on a green table cloth (folds, creases, reflections) surrounded by brown cardboard boxes (damage, imprints, tapes), which exaggerated the sim2real gap. Additionally, the camera model in simulation (Raspberry Pi Camera V2) and reality (Intel RealSense D435i) were different, resulting in a perception gap due to film grain, exposure, resolution, and field of view (FOV) of the perceived image.

[1] **Website:** `https://autodrive-ecosystem.github.io`


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Fig. 5: Results of the 1st ablation study performed on (a) sim and (b) real camera frames to assess the effect of domain adaptation direction {(c) sim2real, (d) real2sim}, foundation model {(e) SDXL, (f) SDXL-Turbo}, denoising steps {(g) 1, (h) 5, (i) 30}, and input resolution {(j) 2560×1096, (k) 1280×548, (l) 640×274}.

4) Evaluation Metrics: The following metrics were used to assess the quality and performance of the proposed sim2real diffusion framework:

- **Feature Similarity:** Feature similarity is evaluated by encoding the input and output images using the CLIP-ViT-L/14 encoder and computing the cosine similarity (CS) of the resulting embedding vectors.

$$\begin{aligned}\mathrm{CS}(A,B) = \cos(\theta) &= \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\,\|\mathbf{B}\|} \\ &= \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\cdot\sqrt{\sum_{i=1}^{n} B_i^2}} \qquad (7)\end{aligned}$$

  Cosine similarity ranges from -1 (perfect dissimilarity) to 1 (perfect similarity), with 0 indicating no similarity. It focuses on the relative pattern of features (e.g., shapes, textures, or objects) and ignores anything that only affects the magnitude of the vectors (e.g., lighting or contrast).
- **Context Similarity:** CLIP directional similarity (CLIP-DS) evaluates the contextual alignment between the {input image, original caption} and {output image, modified caption}. It maps both images and captions into a common high-dimensional space, where similar concepts are close together, and computes the cosine similarity of the resulting embedding vectors. This metric ranges from -1 (perfect dissimilarity) to 1 (perfect similarity) and focuses on the relative pattern of contextual features (e.g., shapes or textures in images, and words or phrases in text).
- **Content Difference:** Content difference is evaluated using learned perceptual image patch similarity (LPIPS) [75], which is a deep-learning-based perceptual metric that measures how perceptually similar two images are by computing the L_2 distance between their feature activations at multiple layers of AlexNet [76].

$$\begin{aligned}\mathrm{LPIPS}(x,y) = \sum_{l} \frac{1}{H_l\cdot W_l} \sum_{h,w} \\ \left\| w_l \odot \left(\phi^l(x)_{h,w} - \phi^l(y)_{h,w}\right)\right\|_2^2 \qquad (8)\end{aligned}$$

  LPIPS score ranges from 0 (perfect similarity) to 1 (completely different). It evaluates perceptual quality at a high level (texture, details, global image structure).
- **Style Difference:** Neural style loss is evaluated using Gram matrix difference, which measures the difference in texture and style between two images. It is computed as the mean squared error between Gram matrices of feature maps (typically from VGG [77] layers).

$$\mathrm{SD}(x,s) \;=\; \sum_{l\in L} w_l \left\| \frac{F_x^l (F_x^l)^\top}{C_l H_l W_l} - \frac{F_s^l (F_s^l)^\top}{C_l H_l W_l} \right\|_F^2 \qquad (9)$$

  We compute the style difference between (i) input-style pair (SD-IS), and (ii) output-style pair (SD-OS), in order to establish the relative reduction in style difference.
- **Performance:** Iterations per second (IPS) is a performance measure to assess the denoising rate. It measures the number of denoising steps $\mathrm{n_{iter}}$ completed within a predefined time interval ΔT.

$$\mathrm{IPS} = \frac{\mathrm{n_{iter}}}{\Delta T} \qquad (10)$$

  IPS values can range from 0 (no update) to upwards of 6 (sufficient for low-speed applications), with >30 IPS being excellent for general-purpose autonomy.

III. Results and Discussion

A. Ablation Studies

The first ablation study (refer Fig. 5) was meant to assess the fitness of latent diffusion models for domain adaptation. Particularly, a common set of virtual and real camera frames (from Nigel dataset) was provided to the diffusion model(s) to assess their capabilities across a range of experiments:

- **Domain Adaptation Direction:** It was observed that adapting real-world images to simulated images (a.k.a. real2sim mapping) was semantically better than sim2real mapping. This can be attributed to the fact that mapping a higher complexity domain onto a lower one is easier. However, being a many-to-one mapping, real2sim domain adaptation can face scalability issues across varied ODDs.
- **Foundation Model:** It was observed that for the common task of sim2real mapping, the base model (SDXL) performs much better than its corresponding time-distilled counterpart (SDXL-Turbo). This can attributed to the fact that time-distillation usually causes over-creativity, which is not suitable for domain adaptation.

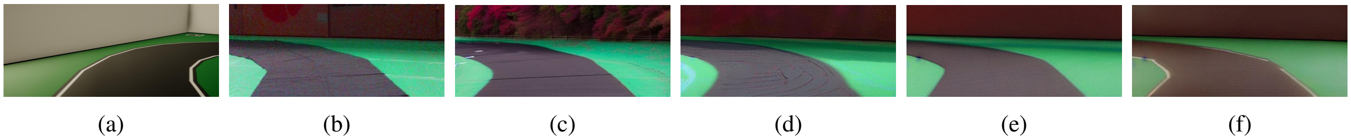


Fig. 6: Results of the 2nd ablation study performed on model architecture to assess the effect of image-prompting {(c, d) no, (e, f) yes} and fine-tuning {(c, e) no, (d, f) yes}, for a common (b) input image and (a) image prompt (if applicable).

- **Denoising Steps:** It was observed that for the common task of real2sim mapping using the same model (SDXL-Turbo), denoising steps heavily influence the generative output. While too few steps (e.g., just 1) lead to noisy output with insufficient domain adaptation, too many of them (e.g., more than 30) quickly lead to hallucination (notice the basket ball, hoop, and players generated by the model in Fig. 5(i)). Each model has its own sweet spot, depending on the task.
- **Input Resolution:** While the proposed pipeline is agnostic to the input image size/resolution, these values do affect the generative process. It was observed that for the common task of sim2real mapping using the same base model (SDXL), reducing the input resolution progressively degraded the generative output, with a significant feature loss beyond a certain threshold (640×274 px, refer Fig. 5(l)). It was later observed that fine-tuning could adapt the foundation model(s) to novel input resolutions.

The second ablation study (refer Fig. 6 and Table I) was performed to assess the effects of fine-tuning and image-prompting on latent diffusion models. Similar to the earlier study, a common set of input (real) and prompt (sim) frames were provided to the diffusion model(s) to assess their domain adaptation capabilities across a range of experiments:

- **Image-Prompting:** Image prompt (IP) provides additional conditioning, which significantly aids in domain adaptation (compare Fig. 6(c) with Fig. 6(d), or Fig. 6(e) with Fig. 6(f)). It was observed that, as opposed to the vanilla models, those conditioned via an image prompt performed better across all the quantitative metrics (compare Table I rows 1 and 2, or 3 and 4), albeit adding a small processing overhead.
- **Fine-Tuning:** Fine-tuning teaches the foundation model a custom concept, which serves a dual purpose. Firstly, it improves the qualitative (compare Fig. 6(c) with Fig. 6(e), or Fig. 6(d) with Fig. 6(f)) and quantitative (compare Table I rows 1 and 3, or 2 and 4) performance of domain adaptation. Secondly, fine-tuning using a *"trigger word"* for each concept makes this approach highly modular and scalable across novel domain gaps. Finally, combining fine-tuning with image-prompting usually results in the most effective domain adaptation, as marked by the highest (44.41%) reduction in style difference.

The third ablation study served the purpose of assessing the scalability/generalizability across exaggerated domain gaps, without image prompting. To this end, the latent diffusion model fine-tuned on 3 different concepts (on-road, racing, and off-road) was stress-tested for its creative/generative capabilities. Fig. 7 shows that the fine-tuned model is able to map domains across various weathers, seasons, and times of the day, and is also capable of mapping one ODD to another (refer Fig. 7(g) or Fig. 7(j)) or altering the ODD altogether (refer Fig. 7(i)), simply using text prompts.

TABLE I: Ablation Study on Model Architecture

Model	**CS ↑**	**CLIP-DS ↑**	**LPIPS ↓**	**SD-OS† ↓**	**IPS‡ ↑**
BM	7.04e-01	6.10e-02	3.74e-01	9.17e-04	7.36
BM+IP	7.58e-01	1.17e-01	3.43e-01	7.80e-04	6.73
FM	8.05e-01	1.05e-01	**2.64e-01**	7.53e-04	**7.42**
FM+IP	**8.19e-01**	**1.78e-01**	2.76e-01	**6.89e-04**	6.86

†SD-IS = 1.24e-03, ‡Laptop GPU: 3080 Ti.

As a natural extension to the third ablation study, we analyzed the effect of image prompt on fine-tuned model in the fourth ablation study. The focus of this study was to assess the semantic and geometric consistency during domain adaptation, conditioned via text as well as image prompts. It is to be noted that the prompt image was randomly sampled from the pool of a few simulation images, and had no paired relationship with the input image (which was sampled from the real-world data). Results of this ablation study (refer Fig. 8) show that the model preserves semantic and geometric features during domain adaptation, while ensuring adequate style transfer. Some of the prominent highlights from the on-road domain adaptation (Fig. 8(a-c)) include generation of road lane markings, smoothing of the creases/reflections on the green table cloth and black tissues, removing box boundaries, toning the color of boxes closer to whiter shades, and removal of any graphical imprints on the boxes. For the racing domain adaptation (Fig. 8(d-f)), some of the noticeable adaptations include smoothing ground and duct textures, preserving original vehicle and environment features, attenuating reflections/glare, and color tone-mapping (e.g., changing the color of LIDAR cap from orange to yellow). Finally, for the off-road domain adaptation (Fig. 8(g-i)), we can notice that the model has preserved the path geometry and semantics, mapped the sidewalk pavement to a dirt road texture, adapted the mowed lawn to tall dry grass, and replaced the real trees with simulated ones.

B. Performance Study

The performance evaluation study (refer Table II) captures training time as well as inference speed across 6 different compute settings (1 local resource, 1 Google Colab resource, and 4 Palmetto Cluster resources).

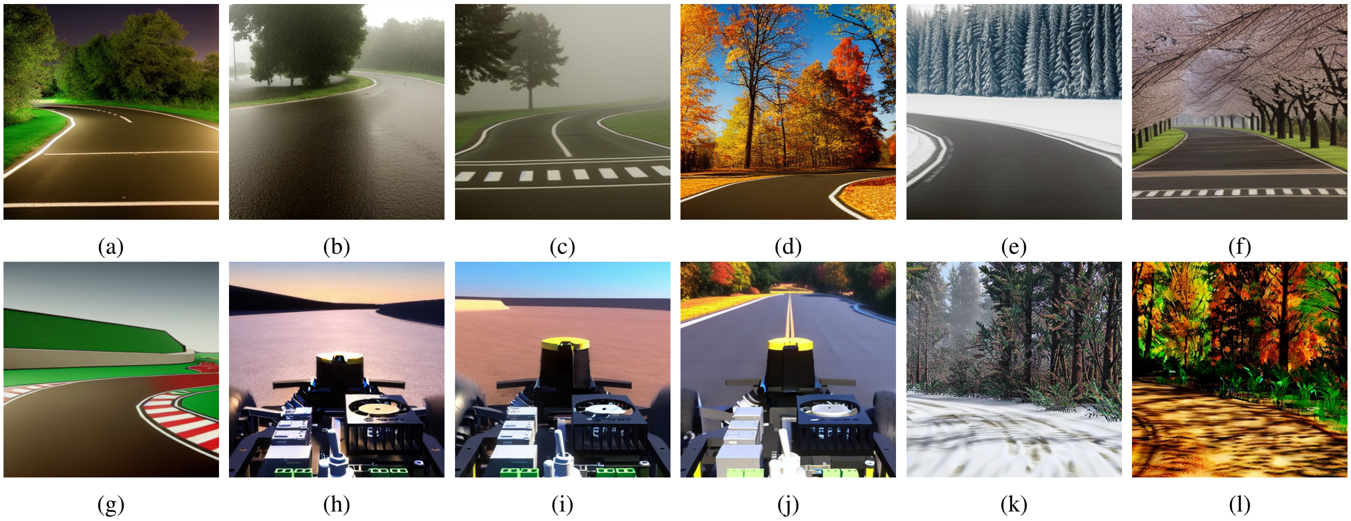


Fig. 7: Results of the 3rd ablation study performed on fine-tuned model without image prompting to assess the domain adaptation capabilities: <autodrive_small_onroad> (a) at night, (b) in rain, (c) in fog, (d) during fall, (e) during winter, (f) during spring, (g) on racetrack; <autodrive_small_racing> (h) at sunrise, (i) in desert, (j) on public road; <autodrive_large_offroad> (k) in snow, (l) during fall.

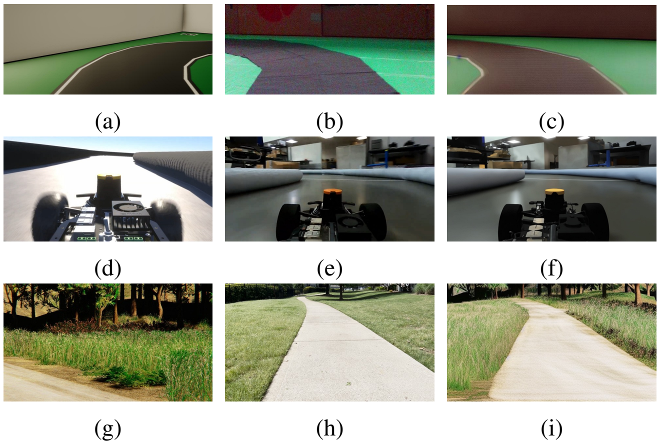


Fig. 8: Results of the 4th ablation study performed on fine-tuned model with image prompting to qualitatively assess the domain adaptation capabilities: <autodrive_small_onroad> (a) prompt, (b) input, (c) output; <autodrive_small_racing> (d) prompt, (e) input, (f) output; <autodrive_large_offroad> (g) prompt, (h) input, (i) output.

TABLE II: Performance Evaluation of Sim2Real Diffusion

Performance Metric	**3080 Ti**	**T4**	**V100**	**A100**	**H100**	**L40S**
Training Time (mm:ss)	13:33	21:42	05:12	02:58	02:37	02:26
Inference Speed (IPS)	07.07	03.76	23.43	32.52	49.69	65.07

TABLE III: Quantitative Evaluation of Sim2Real Diffusion

Statistic	**CS** ↑	**CLIP-DS** ↑	**LPIPS** ↓	**SD-OS**† ↓	**IPS**‡ ↑
Best (min/max)	9.01e-01	2.09e-01	2.42e-01	6.32e-04	7.69
Mean (μ)	8.15e-01	1.08e-01	4.06e-01	7.89e-04	7.07
Std. Dev. (σ)	4.06e-02	3.47e-02	7.22e-02	1.15e-04	0.23

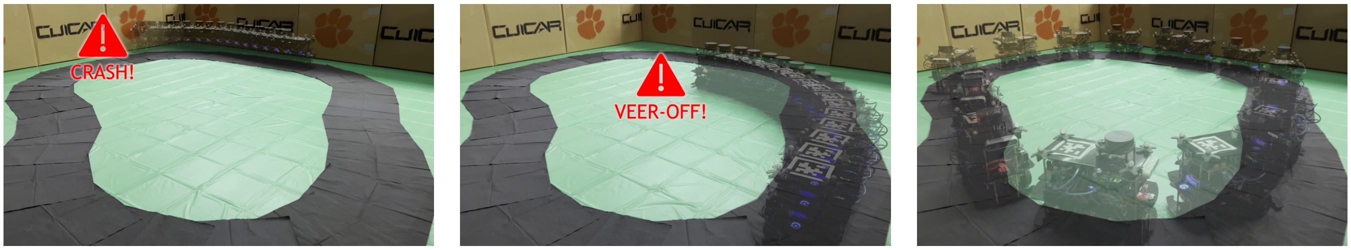
†SD-IS = {μ: 1.32e-03, σ: 1.31e-04}, ‡Laptop GPU: 3080 Ti.

We can observe that fine-tuning the foundation models using the proposed pipeline is possible across all computing platforms. The worst-case training time reported is just a little over 20 minutes (T4 GPU via Google Colab Free Tier), which ensures that the framework is serviceable to others in the community, who potentially do not have access to high-performance computing resources. Additionally, compared to the training time for the behavioral cloning (>1.5 hours) or deep reinforcement learning (DRL) policies (typically >10 hours), fine-tuning the diffusion model is relatively trivial. Additionally, the fine-tuning process requires only a handful (5-10) domain-specific examples to learn the new concept. This ensures that the proposed pipeline is not only fast but also highly scalable across different domain representations.

It is also worth noting that the model can provide pseudo-real-time inference even on local compute resources. As highlighted earlier, slow-speed autonomous systems (e.g., Nigel in this case) can run sim2real diffusion to complete their objective in real-time. Additionally, distributed computing frameworks can alleviate any edge-computing limitations (memory, throughput, etc.), potentially supporting faster-than-real-time inference (e.g., >65 IPS with L40S GPU).

C. Case Study

The exemplar case study serves the purpose of demonstrating the serviceability of the proposed sim2real transfer method (refer Fig. 9 and Table III). It is worth mentioning that while earlier work [67] has demonstrated zero-shot transferability of behavioral cloning, we deliberately exaggerated the sim2real gap by choosing a dissimilar real-world setup and pruning the data augmentation step. This allowed us to demonstrate sim2real diffusion during deployment, while also reducing the data requirement for behavioral cloning by $\sim 64\times$ (translating to $> 50\times$ training time reduction) without compromising on the sim2real transfer.



(a) (b) (c)

Fig. 9: Results of sim2real transfer of the end-to-end driving algorithm trained using simulation-only data without any augmentations: (a, b) exemplar instances where the algorithm fails to traverse across the road (with possibility for collisions) when sim2real diffusion is OFF, and (c) successful autonomous driving in the real world when sim2real diffusion is ON.

When sim2real diffusion is turned off, the model frequently fails to navigate the road properly, as illustrated in Fig. 9(a-b), where the vehicle veers off the course, potentially leading to collisions. These failures are primarily attributed to the distributional shift between the synthetic training domain and the complexities of the real-world domain (i.e., sim2real gap). Without adaptation, the model overfits to simulation-specific artifacts and fails to learn representations that are robust to real-world variations such as changes in lighting, textures, colors, reflections, glare, sensor noise, etc.

In contrast, enabling sim2real diffusion significantly mitigates these issues by learning a more transferable representation. As shown in Fig. 9(c), the same behavioral cloning model, when deployed with sim2real diffusion enabled, is capable of executing successful and stable autonomous driving maneuvers under real-world conditions. This improvement can be attributed to the diffusion process aligning the synthetic and real data distributions through learned image-level transformations and feature-space adaptation, thereby reducing the domain discrepancy at both the input and intermediate representation levels. These empirical results support the hypothesis that sim2real diffusion acts as an effective domain adaptation strategy (40.33% reduction in style difference), allowing reliable policy transfer from simulation to reality.

IV. Conclusion

In this work, we addressed some of the limitations of existing sim2real transfer approaches for autonomous driving by proposing a unified framework based on conditional latent diffusion models. Our proposed framework specifically targets autonomy-oriented requirements that have remained underexplored in prior methods -— namely, the need for conditioned domain adaptation, few-shot generalization, modular handling of multiple domain shifts while remaining scalable and real-time executable. Through extensive experiments and ablation studies, we demonstrated the efficacy of our approach in bridging the perceptual domain gap between simulated and real-world driving environments. Notably, we observed over 40% improvement in bridging the perceptual sim2real gap in our case study (end-to-end behavioral cloning for autonomous driving). These findings highlight the importance of leveraging generative diffusion models as a flexible and scalable solution to sim2real transfer challenges.

Future research could attempt to further improve the real-time performance of the proposed framework via time or knowledge distillation techniques. Additionally, we wish to expand the existing framework to learn more simulation styles across different vehicles, environments, and ODDs. Finally, incorporating custom guardrails and physics-based machine learning principles can potentially improve the domain adaptation capabilities while offering trustworthy grounding.

References

[1] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, C. Li, F. Meier, D. Negrut, L. Righetti, A. Rodriguez, J. Tan, and J. Trinkle, "On the Use of Simulation in Robotics: Opportunities, Challenges, and Suggestions for Moving Forward," *Proceedings of the National Academy of Sciences*, vol. 118, no. 1, p. e1907856118, 2021. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1907856118

[2] Samak, Tanmay, Samak, Chinmay, Krovi, Venkat, Binz, Joey, Luo, Feng, Smereka, Jonathon, Brudnak, Mark, and Gorsich, David, "Off-Road Autonomy Validation Using Scalable Digital Twin Simulations Within High-Performance Computing Clusters," in *2024 NDIA Michigan Chapter Ground Vehicle Systems Engineering and Technology Symposium*. National Defense Industrial Association, sep 2024. [Online]. Available: https://doi.org/10.4271/2024-01-4111

[3] Kagalwala, Huzefa, Srivastava, Siddhant, Venkatesan, Manikanda Balaji, Srinivasan, Srivatsan, and Krovi, Venkat N, "Implementation Methodologies for Simulation as a Service (SaaS) to Develop ADAS Applications," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 3, no. 4, pp. 2123–2135, apr 2021. [Online]. Available: https://doi.org/10.4271/2021-01-0116

[4] Y. Koroglu and F. Wotawa, "Towards a Review on Simulated ADAS/AD Testing," in *2023 IEEE/ACM International Conference on Automation of Software Test (AST)*, 2023, pp. 112–122.

[5] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner Case Generation and Analysis for Safety Assessment of Autonomous Vehicles," *Transportation Research Record*, vol. 2675, no. 11, pp. 587–600, 2021. [Online]. Available: https://doi.org/10.1177/03611981211018697

[6] e. a. Feng, S., "Intelligent Driving Intelligence Test for Autonomous Vehicles with Naturalistic and Adversarial Environment," *Nat Commun*, vol. 12, no. 1, 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-21007-8

[7] S. Guneshka, "Ontology-Based Corner Case Scenario Simulation for Autonomous Driving," 2022.

[8] Z. Song, Z. He, X. Li, Q. Ma, R. Ming, Z. Mao, H. Pei, L. Peng, J. Hu, D. Yao, and Y. Zhang, "Synthetic Datasets for Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, p. 1847–1864, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1109/TIV.2023.3331024

[9] D. Liu, Y. Wang, K. E. Ho, Z. Chu, and E. Matson, "Virtual World Bridges the Real Challenge: Automated Data Generation for Autonomous Driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 159–164.

[10] M. Goyal and Q. H. Mahmoud, “A Systematic Review of Synthetic Data Generation Techniques Using Generative AI,” *Electronics*, vol. 13, no. 17, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/17/3509

[11] V. B. Cecchetti, B. J. Souza, and R. Z. Freire, “Framework for Automated Synthetic Image Generation for Vehicle Detection,” in *Proceedings of the 2023 6th International Conference on Sensors, Signal and Image Processing*, ser. SSIP ’23. New York, NY, USA: Association for Computing Machinery, 2024, p. 8–13. [Online]. Available: https://doi.org/10.1145/3653863.3653872

[12] M. Cao and R. Ramezani, “Data Generation Using Simulation Technology to Improve Perception Mechanism of Autonomous Vehicles,” *Journal of Physics: Conference Series*, vol. 2547, no. 1, p. 012006, jul 2023. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2547/1/012006

[13] T. Zhang, H. Liu, W. Wang, and X. Wang, “Virtual Tools for Testing Autonomous Driving: A Survey and Benchmark of Simulators, Datasets, and Competitions,” *Electronics*, vol. 13, no. 17, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/17/3486

[14] A. Remonda, N. Hansen, A. Raji, N. Musiu, M. Bertogna, E. Veas, and X. Wang, “A Simulation Benchmark for Autonomous Racing with Large-Scale Human Data,” 2024. [Online]. Available: https://arxiv.org/abs/2407.16680

[15] D. Paz, P.-j. Lai, N. Chan, Y. Jiang, and H. I. Christensen, “Autonomous Vehicle Benchmarking using Unbiased Metrics,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6223–6228.

[16] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. D. Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, S. Legg, V. Mnih, K. Kavukcuoglu, and D. Silver, “Massively Parallel Methods for Deep Reinforcement Learning,” 2015. [Online]. Available: https://arxiv.org/abs/1507.04296

[17] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, “Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning,” 2021. [Online]. Available: https://arxiv.org/abs/2108.10470

[18] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning,” in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 91–100. [Online]. Available: https://proceedings.mlr.press/v164/rudin22a.html

[19] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson Surface Reconstruction,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP ’06. Goslar, DEU: Eurographics Association, 2006, p. 61–70.

[20] C. V. Samak, T. V. Samak, A. Joglekar, U. Vaidya, and V. Krovi, “Digital Twins Meet the Koopman Operator: Data-Driven Learning for Robust Autonomy,” 2024. [Online]. Available: https://arxiv.org/abs/2409.10347

[21] J. L. Schönberger and J.-M. Frahm, “Structure-from-Motion Revisited,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[22] L. Pan, D. Baráth, M. Pollefeys, and J. L. Schönberger, “Global Structure-from-Motion Revisited,” 2024. [Online]. Available: https://arxiv.org/abs/2407.20219

[23] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” *Commun. ACM*, vol. 65, no. 1, p. 99–106, Dec. 2021. [Online]. Available: https://doi.org/10.1145/3503250

[24] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, “Block-NeRF: Scalable Large Scene Neural View Synthesis,” 2022. [Online]. Available: https://arxiv.org/abs/2202.05263

[25] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3D Gaussian Splatting for Real-Time Radiance Field Rendering,” 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[26] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, “Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction,” 2023. [Online]. Available: https://arxiv.org/abs/2309.13101

[27] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang, “Periodic Vibration Gaussian: Dynamic Urban Scene Reconstruction and Real-time Rendering,” 2024. [Online]. Available: https://arxiv.org/abs/2311.18561

[28] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, “Street Gaussians: Modeling Dynamic Urban Scenes with Gaussian Splatting,” 2024. [Online]. Available: https://arxiv.org/abs/2401.01339

[29] J. Kim and C. Park, “End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1194–1202.

[30] S. Akhauri, L. Y. Zheng, and M. C. Lin, “Enhanced Transfer Learning for Autonomous Driving with Systematic Accident Simulation,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5986–5993.

[31] C. Wu, X. Bi, J. Pfrommer, A. Cebulla, S. Mangold, and J. Beyerer, “Sim2real Transfer Learning for Point Cloud Segmentation: An Industrial Application Case on Autonomous Disassembly,” in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 4520–4529.

[32] Y. Shukla, C. Thierauf, R. Hosseini, G. Tatiya, and J. Sinapov, “ACuTE: Automatic Curriculum Transfer from Simple to Complex Environments,” in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS ’22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2022, p. 1192–1200.

[33] L. Væhrens, D. D. Álvarez, U. Berger, and S. Bøgh, “Learning Task-Independent Joint Control for Robotic Manipulators with Reinforcement Learning and Curriculum Learning,” in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022, pp. 1250–1257.

[34] C. Xiao, P. Lu, and Q. He, “Flying Through a Narrow Gap Using End-to-End Deep Reinforcement Learning Augmented With Curriculum Learning and Sim2Real,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, pp. 2701–2708, 2023.

[35] B. Qin, Y. Gao, and Y. Bai, “Sim-to-Real: Six-legged Robot Control with Deep Reinforcement Learning and Curriculum Learning,” in *2019 4th International Conference on Robotics and Automation Engineering (ICRAE)*, 2019, pp. 1–5.

[36] I. Clavera, A. Nagabandi, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, “Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HyztsoC5Y7

[37] Y. Jaafra, A. Deruyver, J. L. Laurent, and M. S. Naceur, “Context-Aware Autonomous Driving Using Meta-Reinforcement Learning,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 450–455.

[38] A. Kar, A. Prakash, M.-Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, and S. Fidler, “Meta-Sim: Learning to Generate Synthetic Datasets,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4550–4559.

[39] K. Arndt, M. Hazara, A. Ghadirzadeh, and V. Kyrki, “Meta Reinforcement Learning for Sim-to-Real Domain Adaptation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2725–2731.

[40] C. Finn, P. Abbeel, and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135. [Online]. Available: https://proceedings.mlr.press/v70/finn17a.html

[41] Anonymous, “Meta-Reinforcement Learning for Adaptive Autonomous Driving,” in *ICML Workshop on Adaptive & Multitask Learning: Algorithms & Systems*, 2019. [Online]. Available: https://openreview.net/forum?id=S1eoN9rsnN

[42] X. Chen, K. Chen, M. Zhu, H. F. Yang, S. Shen, X. Wang, and Y. Wang, “MetaFollower: Adaptable Personalized Autonomous Car Following,” *Transportation Research Part C: Emerging Technologies*, vol. 169, p. 104872, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X24003930

[43] Y. Tsuchiya, T. Balch, P. Drews, and G. Rosman, “Online Adaptation of Learned Vehicle Dynamics Model with Meta-Learning Approach,” 2024. [Online]. Available: https://arxiv.org/abs/2409.14950

[44] M. R. U. Saputra, P. Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni, “Distilling Knowledge From a Deep Pose Regressor

Network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 263–272.

[45] L. Zhang, R. Dong, H.-S. Tai, and K. Ma, "PointDistiller: Structured Knowledge Distillation Towards Efficient and Compact 3D Detection," 2022. [Online]. Available: https://arxiv.org/abs/2205.11098

[46] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet, "Image-to-Lidar Self-Supervised Distillation for Autonomous Driving Data," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 9881–9891.

[47] J. Li, H. Dai, and Y. Ding, "Self-Distillation for Robust LiDAR Semantic Segmentation in Autonomous Driving," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 659–676.

[48] M. Malmir, J. Josifovski, N. Klarmann, and A. Knoll, "DiAReL: Reinforcement Learning with Disturbance Awareness for Robust Sim2Real Policy Transfer in Robot Control," 2023. [Online]. Available: https://arxiv.org/abs/2306.09010

[49] J. W. Kim, H. Shim, and I. Yang, "On Improving the Robustness of Reinforcement Learning-Based Controllers using Disturbance Observer," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 847–852.

[50] J. Josifovski, M. Malmir, N. Klarmann, B. L. Žagar, N. Navarro-Guerrero, and A. Knoll, "Analysis of Randomization Effects on Sim2Real Transfer in Reinforcement Learning for Robotic Manipulation Tasks," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 10 193–10 200.

[51] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization Without Accessing Target Domain Data," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2100–2110.

[52] G. D. Kontes, D. D. Scherer, T. Nisslbeck, J. Fischer, and C. Mutschler, "High-Speed Collision Avoidance using Deep Reinforcement Learning and Domain Randomization for Autonomous Vehicles," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.

[53] S. Pouyanfar, M. Saleem, N. George, and S.-C. Chen, "ROADS: Randomization for Obstacle Avoidance and Driving in Simulation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1267–1276.

[54] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to Drive from Simulation without Real World Labels," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4818–4824.

[55] C. Y. Zhang and A. Shrivastava, "AptSim2Real: Approximately-Paired Sim-to-Real Image Translation," 2023. [Online]. Available: https://arxiv.org/abs/2303.12704

[56] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-Consistent Adversarial Domain Adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1989–1998. [Online]. Available: https://proceedings.mlr.press/v80/hoffman18a.html

[57] J. Zhang, L. Tai, P. Yun, Y. Xiong, M. Liu, J. Boedecker, and W. Burgard, "VR-Goggles for Robots: Real-to-Sim Domain Adaptation for Visual Control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1148–1155, 2019.

[58] S. Tripathy, J. Kannala, and E. Rahtu, "Learning Image-to-Image Translation Using Paired and Unpaired Training Samples," in *Computer Vision – ACCV 2018*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 51–66.

[59] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 674–10 685.

[60] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[61] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[62] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[64] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models," 2023. [Online]. Available: https://arxiv.org/abs/2308.06721

[65] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[66] G. Ilharco, M. Wortsman, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, "OpenCLIP," July 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5143773

[67] T. Samak, C. Samak, S. Kandhasamy, V. Krovi, and M. Xie, "AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education," *Robotics*, vol. 12, no. 3, p. 77, May 2023. [Online]. Available: http://dx.doi.org/10.3390/robotics12030077

[68] T. V. Samak and C. V. Samak, "AutoDRIVE - Technical Report," 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2211.08475

[69] T. V. Samak, C. V. Samak, and M. Xie, "AutoDRIVE Simulator: A Simulator for Scaled Autonomous Vehicle Research and Education," in *2021 2nd International Conference on Control, Robotics and Intelligent System*, ser. CCRIS'21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1–5. [Online]. Available: https://doi.org/10.1145/3483845.3483846

[70] T. V. Samak and C. V. Samak, "AutoDRIVE Simulator - Technical Report," 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2211.07022

[71] T. V. Samak, C. V. Samak, and V. N. Krovi, "Towards Validation of Autonomous Vehicles Across Scales using an Integrated Digital Twin Framework," in *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2024, pp. 1068–1075.

[72] C. V. Samak, T. V. Samak, J. M. Velni, and V. N. Krovi, "Nigel — Mechatronic Design and Robust Sim2Real Control of an Overactuated Autonomous Vehicle," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 4, pp. 2785–2793, 2024.

[73] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 22 500–22 510.

[74] T. V. Samak, C. V. Samak, and S. Kandhasamy, "Robust Behavioral Cloning for Autonomous Vehicles Using End-to-End Imitation Learning," *SAE International Journal of Connected and Automated Vehicles*, vol. 4, no. 3, pp. 279–295, August 2021. [Online]. Available: https://doi.org/10.4271/12-04-03-0023

[75] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[76] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[77] S. Liu and W. Deng, "Very Deep Convolutional Neural Network Based Image Classification using Small Training Sample Size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015, pp. 730–734.