Vietnam National University-Ho Chi Minh City

Ho Chi Minh University of Science

Faculty of Information Tecnology

Knowledge Engineering Department

HyperNeRFGAN: Hypernetwork approach to 3D NeRF GAN

Course: Introduction to Machine Learning

Student Group:
23127165 - Nguyen Hai Dang
23127438 - Dang Truong Nguyen

Instructors:
Mr. Bui Duy Dang
Mr. Huynh Lam Hai Dang
Mr. Tran Trung Kien

January 4, 2026

Contents

1 Introduction

1.1 Group's Information and Contribution

- Course name: Introduction to Machine Learning
- Supervising lecturer: Mr Huynh Lam Hai Dang and Mr Tran Trung Kien
- Research paper name: HyperNeRFGAN: Hypernetwork approach to 3D NeRF GAN
- Duration: From October 29, 2025 to December 28, 2025
- Team members and contributions:

No.	Student ID	Full Name	Role	Contributions	Percentage
01	23127165	Nguyen Hai Dang	Team Leader	- Project planning and coordination; - Literature review (NeRF, GANs, Hypernetworks); - HyperNeRFGAN architecture understanding; - Low-memory optimization for Tesla T4; - Ablation study design and implementation; - Main report writing (Methodology, Results, Discussion)	100%
02	23127438	Dang Truong Nguyen	Member	- Codebase setup and environment configuration; - Dataset preparation and I/O optimization; - Training execution and experiment monitoring; - Result visualization and figure preparation; - Report writing (Implementation Details, Conclusion)	100%

Table 1: Team members and individual contributions

1.2 Overall Completion

Requirement Completion	Percentage (%)
Comprehensive report with theoretical background and methodology description	100%
Reproduction of HyperNeRFGAN architecture (Hypernetwork + NeRF + GAN)	100%
Successful training on the CARLA and ShapeNet (Cars, Chairs, Planes) dataset under constrained hardware (Tesla T4)	100%
Implementation of low-memory optimization strategies (patch size, ray sampling, accumulation)	100%
Qualitative evaluation including latent space interpolation	100%
Ablation studies on sampling density, patch size, regularization, and augmentation	100%
Experimental analysis and interpretation of failure modes	100%
Discussion of architectural trade-offs and memory limitations	100%
Clear conclusions and future work directions	100%

Abstract

Recent advancements in 3D-aware image synthesis have successfully integrated Neural Radiance Fields (NeRF) into Generative Adversarial Networks (GANs). However, standard approaches often suffer from prohibitive computational costs and a reliance on accurate camera pose information, restricting their accessibility. In this report, we present a rigorous reproduction and empirical analysis of **HyperNeRFGAN**, an architecture that leverages hypernetworks to generate view-independent NeRF representations. Operating under strict hardware constraints (a single NVIDIA Tesla T4 GPU with 16GB VRAM), we successfully adapt the training pipeline by implementing memory-aware optimizations, specifically reducing the volumetric sampling density ($N_s = 32$) and rendering patch size ($P_s = 32$).

We evaluate the model's generalizability across four distinct datasets—**CARLA**, **ShapeNet Cars**, **Planes**, and **Chairs**—demonstrating consistent convergence and geometric plausibility within a limited training budget of **200 kimg**. Furthermore, we conduct a comprehensive ablation study on the CARLA dataset to quantify the sensitivity of the hypernetwork paradigm. Our results establish that volumetric sampling density is irreducible, with sparse sampling ($N_s = 8$) causing catastrophic geometric collapse, while standard GAN stabilization techniques (adaptive augmentation and R_1 regularization) remain essential to prevent mode collapse and training instability. These findings validate HyperNeRFGAN as a viable solution for 3D generative modeling on consumer-grade hardware, provided that rendering budgets are carefully balanced against geometric fidelity.

2 Background & Motivation

The synthesis of high-fidelity three-dimensional objects from unstructured 2D image collections remains a pivotal challenge in computer vision. While Generative Adversarial Networks (GANs) have demonstrated remarkable success in 2D image synthesis, extending these capabilities to the 3D domain requires the incorporation of explicit geometric inductive biases. Neural Radiance Fields (NeRF) have emerged as a de facto standard for this purpose, offering a continuous volumetric representation that enforces multi-view consistency. However, integrating NeRFs into adversarial training pipelines introduces severe computational bottlenecks and optimization challenges, primarily due to the excessive memory demands of volumetric ray marching and the sensitivity of adversarial objectives to geometric inconsistencies.

2.1 Problem Formulation

We address the problem of *unsupervised 3D-aware image synthesis*. Let $\mathcal{D} = \{I_1, I_2, \ldots, I_N\}$ be a dataset of 2D images representing a specific object category (e.g., cars), where each image $I_i \in \mathbb{R}^{H \times W \times 3}$ is captured from an unknown viewpoint. We assume no access to camera pose labels or 3D geometry ground truth during training.

Our goal is to learn a generator function $G : \mathcal{Z} \to \mathcal{I}$ that maps a latent code $z \in \mathcal{Z}$ (sampled from a prior distribution, typically Gaussian $\mathcal{N}(0, \mathbf{I})$) to a synthesized image $\hat{I}$. Crucially, unlike standard 2D GANs, the generation process must respect an underlying 3D representation to ensure geometric consistency across different views. This can be formalized as a composition of a 3D volume generation function V and a differentiable rendering function π:

$$\hat{I} = \pi(V(z, \theta), \xi), \tag{1}$$

where:

- $V(z, \theta)$ represents the 3D scene (e.g., a neural radiance field) parameterized by weights generated from z.
- ξ represents the camera parameters (viewing direction and position), sampled from a prior distribution P_{cam}.
- π is the volumetric rendering operator that projects the 3D scene onto a 2D plane.

The objective is to train G (and consequently the implicit 3D representation V) in an adversarial manner against a discriminator D, such that the distribution of generated images P_g matches the data distribution P_{data}, while maintaining 3D structural coherence.

2.2 Motivation and Challenges

Existing state-of-the-art approaches, such as π-GAN and GRAF, typically condition a static NeRF network on a latent code through concatenation or feature-wise modulation. While effective, these methods often rely on view-dependent radiance formulations, which require accurate camera pose distributions—a requirement that is frequently unmet in real-world or medical datasets. Furthermore, the high computational cost of querying a 3D field for every pixel during adversarial training generally confines these models to large-scale hardware infrastructures (e.g., NVIDIA V100 or A100

GPU clusters), posing a significant barrier to reproducibility and deployment in resource-constrained environments.

To address these limitations, **HyperNeRFGAN** introduces a paradigm shift by employing a **hypernetwork** to directly predict the weights of a simplified, view-independent NeRF representation. This design decouples geometric generation from the rendering viewpoint, theoretically improving robustness when camera pose information is incomplete, noisy, or entirely unavailable.

2.3 Scope of Investigation

This report presents a rigorous reproduction and empirical analysis of HyperNeRFGAN. We specifically focus on the model's adaptability and performance stability across a diverse set of synthetic object classes, including **CARLA** (cars) and **ShapeNet** (cars, planes, chairs), validating its generalizability for 3D-aware image synthesis. Our investigation was conducted under strict hardware constraints (**Single NVIDIA Tesla T4 GPU, 16GB VRAM**), necessitating aggressive memory optimization strategies, including reduced ray sampling density and efficient data handling.

Beyond mere reproduction, we seek to quantify the impact of key hyperparameters on the stability of the 3D-aware adversarial game. We hypothesize that the delicate balance between the discriminator's regularization and the generator's sampling density is critical for avoiding geometric collapse.

2.4 Contributions

Our contributions are summarized as follows:

1. **Resource-Constrained Generalization:** We successfully reproduce the HyperNeRFGAN training pipeline and demonstrate its effectiveness across **multiple datasets (CARLA, ShapeNet Car, ShapeNet Plane, ShapeNet Chair)** in a constrained environment. This involves optimizing the rendering patch size (P_s) and implementing memory-efficient data handling, proving the architecture's viability on consumer-grade hardware for various object types.

2. **Deep Ablation Analysis:** We conduct a comprehensive ablation study on the **CARLA** dataset to isolate the effects of:

   - **Volumetric Sampling:** Comparing standard sampling ($N_s = 32$) against extreme sparse sampling ($N_s = 8$).
   - **Discriminator Receptive Field:** Analyzing global vs. local coherence by varying patch sizes ($P_s = 32$ vs. $P_s = 8$).
   - **Regularization Dynamics:** Evaluating the impact of relaxing the R_1 gradient penalty ($\gamma = 1.0 \rightarrow 0.5$) and **disabling adaptive augmentation** ($p = 0.0$).

3. **Manifold Continuity Verification:** Through latent space interpolation, we qualitatively verify that the hypernetwork learns a continuous and smooth manifold of 3D structures, rather than memorizing discrete training samples.

3 Methodology

3.1 Architectural Design of HyperNeRFGAN

The core innovation of HyperNeRFGAN lies in its dual-component architecture, which is trained end-to-end within an adversarial framework to synthesize 3D-aware images. The model consists of three primary components:

1. **Generator ($\mathcal{G}$): Hypernetwork.** Acting as the generator in the GAN framework, $\mathcal{G}$ is implemented using a StyleGAN2-based backbone. Its primary function is to transform a latent Gaussian noise vector $\mathbf{z} \in \mathbb{R}^{d_z}$ (where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) into the weights of a target neural network. By generating network parameters rather than intermediate features, $\mathcal{G}$ directly controls the implicit 3D scene representation, providing a higher degree of expressiveness compared to conventional conditioning-based approaches.

2. **Target Network (F_θ): Simplified NeRF MLP.** The target network represents the 3D scene as a continuous function. It is implemented as a simplified multilayer perceptron (MLP) parameterized by weights θ generated by $\mathcal{G}$. Mathematically, F_θ maps spatial coordinates $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ to a volumetric density $\sigma \in \mathbb{R}_{\geq 0}$ and an emitted color $\mathbf{c} \in \mathbb{R}^3$ (an RGB vector).
$$F_\theta : \mathbf{x} \in \mathbb{R}^3 \rightarrow (\mathbf{c}, \sigma) \in \mathbb{R}^3 \times \mathbb{R}_{\geq 0} \tag{2}$$
Crucially, unlike standard NeRF, the viewing direction $\mathbf{d} \in \mathbb{S}^2$ is **excluded** from the input space, rendering the model *view-independent*. This design choice significantly reduces the dimensionality of the learning problem.

3. **Discriminator ($\mathcal{D}$).** The discriminator is a standard 2D convolutional network, also derived from StyleGAN2, which distinguishes between real images $I \in \mathbb{R}^{H \times W \times 3}$ from the dataset and synthetic images rendered via volumetric integration from F_θ.

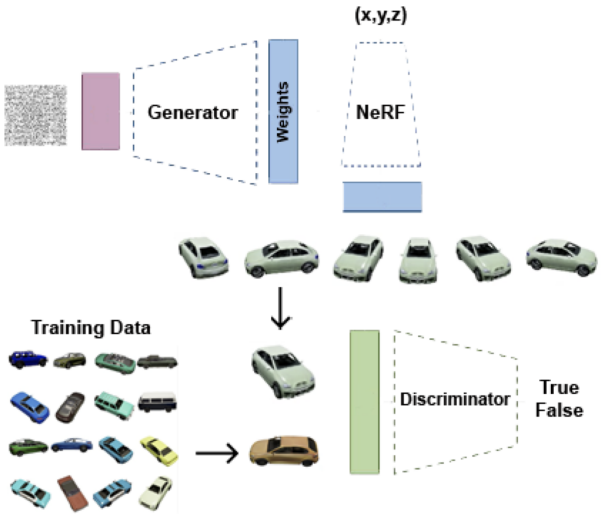


Figure 1: Overview of the HyperNeRFGAN architecture. A latent vector z is mapped by the generator (hypernetwork) to the weights of a view-independent NeRF MLP (F_θ). The NeRF performs volumetric rendering to produce a 2D image, which is evaluated by the discriminator against real samples.

3.2 Theoretical Justification: The Lambertian Assumption

A standard **Neural Radiance Field (NeRF)** approximates the plenoptic function by conditioning the radiance on both spatial position and viewing direction: $\mathbf{c} = f(\mathbf{x}, \mathbf{d})$. This allows modeling view-dependent effects such as specular reflections, where the observed color changes based on the observer's angle ($\mathbf{d}$). However, HyperNeRFGAN simplifies this by adopting the **Lambertian Surface Assumption**.

3.2.1 Definition and Mathematical Implication

A Lambertian surface reflects incident light uniformly in all directions. Consequently, the apparent color of a surface point is isotropic. Mathematically, this implies that the gradient of the color function with respect to the viewing direction is zero: $\nabla_{\mathbf{d}} c(\mathbf{x}, \mathbf{d}) = \mathbf{0}$. Therefore, we can approximate the color function as depending solely on position:

$$c(\mathbf{x}, \mathbf{d}) \approx c(\mathbf{x}). \tag{3}$$

This justification allows us to define the target network F_θ strictly on the domain $\mathbb{R}^3$ rather than $\mathbb{R}^5$.

3.2.2 Why this assumption is critical for the model

The benefits of this assumption are twofold, specifically addressing the challenges of generative modeling on constrained hardware:

1. **Manifold Simplification for the Hypernetwork:** The hypernetwork $\mathcal{G}$ must learn a mapping from the latent space $\mathcal{Z}$ to the functional space of 3D objects. By removing $\mathbf{d}$, we reduce the complexity of the target function from $\mathbb{R}^5 \to \mathbb{R}^4$ to $\mathbb{R}^3 \to \mathbb{R}^4$. Learning to generate weights for "geometry-only" functions (shapes and textures) is significantly more stable than generating weights for functions that must also encode complex physical light interactions.

2. **Stability in Unposed Training:** Since our model is trained via an adversarial process without ground-truth camera poses, retaining view-dependence ($\mathbf{d}$) introduces a critical failure mode. The generator could exploit $\mathbf{d}$ to create "optical illusions"—view-dependent artifacts that look correct from specific 2D angles but fail to form a coherent 3D solid. Enforcing the Lambertian assumption forces the generator to produce consistent geometry to satisfy the discriminator, as it cannot rely on view-dependent tricks.

3.3 Factorized Multiplicative Modulation

Directly predicting all weights of the target NeRF network would impose excessive memory and computational overhead. To address this, HyperNeRFGAN adopts **Factorized Multiplicative Modulation (FMM)**. Rather than generating full weight matrices for the target MLP, the hypernetwork outputs two low-rank modulation matrices, A and B, for each linear layer in F_θ. The effective transformation for a layer with input $\mathbf{h}_{in}$ is defined as:

$$\mathbf{h}_{out} = W \odot (AB) \cdot \mathbf{h}_{in} + b, \tag{4}$$

where W and b denote shared, learnable weights and biases stored in the network, and $\odot$ represents element-wise multiplication. The matrices $A \in \mathbb{R}^{n_{\text{out}} \times k}$ and $B \in \mathbb{R}^{k \times n_{\text{in}}}$ are dynamically generated from z for each sample, with a modulation rank of $k = 10$.

Derivation of Computational Efficiency. The necessity of FMM implies a significant reduction in parameter complexity. Consider a standard linear layer with input dimension n_{in} and output dimension n_{out}. A naive hypernetwork would need to generate a full weight matrix $\hat{W} \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}}$, resulting in a complexity of $\mathcal{O}(n_{\text{out}} \cdot n_{\text{in}})$. By employing FMM, we decompose the modulation into two low-rank matrices. The number of predicted parameters reduces to:

$$\mathcal{O}(k \cdot (n_{\text{out}} + n_{\text{in}})). \tag{5}$$

Given our architecture setting where $n_{\text{in}} = n_{\text{out}} = 128$ and rank $k = 10$, the reduction factor is approximately:

$$\frac{128 \times 128}{10 \times (128 + 128)} = \frac{16384}{2560} \approx 6.4.$$

This rank-k approximation effectively projects the high-dimensional weight space onto a lower-dimensional manifold, making the training of a Hypernetwork feasible on a single Tesla T4 GPU.

3.4 Volumetric Rendering and Discretization

To render a 2D image from the implicit representation F_θ, we employ the classic volume rendering integral derived from the radiative transfer equation. The expected color $C(\mathbf{r})$ of a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ (where $\mathbf{o}$ is the camera origin) is given by:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t))\, dt, \quad \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds\right). \tag{6}$$

Here, $\sigma(\mathbf{r}(t))$ is the volume density at point $\mathbf{r}(t)$, and $T(t)$ represents the accumulated transmittance (probability that the ray travels to t without being blocked).

Numerical Quadrature (Discretization). Since the integral cannot be computed analytically, it is approximated using stratified sampling. The continuous integral is discretized into a Riemann sum over N_s sampled points $\{t_i\}_{i=1}^{N_s}$:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N_s} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \quad \text{with } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right), \tag{7}$$

where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples.

Connection to Failure Modes: This discretization formula provides the mathematical basis for the "cloudy" artifacts observed in our ablation studies (Section 5.3.1) when $N_s = 8$. The term $(1 - \exp(-\sigma_i\delta_i))$ represents the opacity (alpha) of the i-th segment. If N_s is too low, the interval δ_i is large, but the sampling misses the high-density regions of surfaces. According to approximation theory, the error of the Riemann sum scales with the step size δ_i. Consequently, the rendered opacity fails to saturate to 1.0 for solid objects, resulting in semi-transparent, ghost-like geometries.

3.5 Training Objective and Regularization

HyperNeRFGAN is trained using a non-saturating GAN objective with an R_1 gradient penalty to stabilize the adversarial minimax game. The discriminator ($\mathcal{D}$) and generator ($\mathcal{G}$) losses are defined as:

$$\mathcal{L}_\mathcal{D} = \mathbb{E}_{x \sim P_{\text{data}}}[f(D(x))] + \mathbb{E}_{z \sim P_{\text{noise}}}[f(-D(\mathcal{G}(z)))] + R_1(\mathcal{D}), \tag{8}$$

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}_{z \sim P_{\text{noise}}}[f(-D(\mathcal{G}(z)))], \tag{9}$$

where $f(t) = -\log(1 + \exp(-t))$ is the softplus function.

The R_1 regularization term penalizes the squared norm of the discriminator's gradients on real data:

$$R_1(\mathcal{D}) = \frac{\gamma}{2}\mathbb{E}_{x \sim P_{\text{data}}}\left[\|\nabla_x D(x)\|^2\right]. \tag{10}$$

Connection to Lipschitz Continuity. The theoretical motivation for the R_1 penalty is to enforce the discriminator to be a K-Lipschitz continuous function. In adversarial training, if the discriminator D becomes too sharp (i.e., gradients $\|\nabla D\|$ become arbitrarily large), the generator G receives unstable or exploding gradients, leading to divergence. By strictly bounding the rate at which D can change with respect to the input image via the gradient penalty, we ensure a smooth optimization landscape. This is critical for HyperNeRFGAN because the generator must traverse a complex manifold of *network weights*, which is much harder to optimize than standard pixel or feature manifolds.

3.6 Experimental Setup and Ablation Design

All experiments were conducted on a single NVIDIA Tesla T4 GPU (16GB VRAM) on the Google Colab platform.

Datasets: We trained the model on CARLA (10k images) and subsets of ShapeNet (Cars, Planes, Chairs), all processed at 128×128 pixel resolution.

Baseline Configuration (Low-Resource Optimized): To fit the model into 16GB VRAM, we established the following baseline:

- Rendering patch size: $P_s = 32$ (Reduced from 64 to save memory).
- Ray samples per pixel: $N_s = 32$ (Reduced from 64).
- Adaptive Discriminator Augmentation (ADA): Enabled.
- R_1 regularization weight: $\gamma = 1.0$.

Ablation Studies (CARLA): To validate the sensitivity of the hypernetwork, we conducted controlled variations:

1. **Sparse Sampling ($N_s = 32 \to 8$):** Testing the limits of volumetric integration.
2. **Small Receptive Field ($P_s = 32 \to 8$):** Testing the discriminator's global context awareness.
3. **No Augmentation ($p = 0.0$):** Testing data efficiency and overfitting.
4. **Reduced Regularization ($\gamma = 1.0 \to 0.5$):** Testing training stability.

4 Implementation Details

4.1 Hardware and Software Environment

All experiments were conducted on the **Google Colab** platform using a single **NVIDIA Tesla T4 GPU** with 16GB of VRAM. This hardware configuration represents a significantly constrained environment compared to the multi-GPU setups (e.g., NVIDIA V100 or A100 clusters) commonly employed in prior 3D-aware GAN research.

To ensure reproducibility and to resolve dependency conflicts inherent in the original codebase—which relies on legacy StyleGAN2 components—we employed **Micromamba** for lightweight and deterministic environment management. The software stack was pinned to specific versions to guarantee stability:

- **Deep Learning Framework:** PyTorch 1.10.0+cu113 with Torchvision 0.11.0.
- **Build and Packaging Tools:** `pip==23.3.1` and `setuptools<59.6.0` were explicitly enforced to avoid incompatibilities introduced by recent Python packaging standards (PEP 517/660), which affect the compilation of custom CUDA extensions used in StyleGAN2.
- **Configuration Management:** Hydra 1.0.6 and OmegaConf 2.0.6.

4.2 Dataset Preparation and I/O Optimization

To demonstrate the versatility of the model, we utilized four distinct datasets, all processed at a fixed resolution of 128×128 pixels:

- **CARLA:** A dataset consisting of 10,000 synthetic images of vehicles rendered under diverse appearances and viewpoints.
- **ShapeNet Cars, Planes, and Chairs:** Three separate subsets from the ShapeNet repository, representing distinct object categories with varying geometric complexities.

To alleviate the substantial I/O overhead associated with loading large numbers of small image files from Google Drive, we implemented a local caching strategy. All datasets were stored as compressed archives and extracted directly into the Colab instance's ephemeral local storage (`/content/`) prior to training. This approach significantly reduced data loading latency and ensured that GPU memory and compute, rather than I/O throughput, remained the primary performance bottleneck during training.

4.3 Resource-Aware Optimization Strategies

The primary challenge in reproducing HyperNeRFGAN on a Tesla T4 GPU arises from the substantial memory footprint of volumetric rendering. In a standard configuration—rendering 64×64 patches with 64 depth samples per ray—the model must compute gradients for over $64^2 \times 64 \approx 262{,}000$ volumetric samples per forward pass, exceeding the available 16GB VRAM capacity.

To address this limitation across all datasets (CARLA and ShapeNet), we designed a **low-memory baseline configuration** tailored to the T4 GPU:

1. **Patch Size Reduction:** The rendering patch size (P_s) for both the generator and discriminator was reduced from the default 64×64 to 32×32 pixels. This modification reduces the number of rays—and consequently the memory required for ray marching—by a factor of four.

2. **Sampling Density Adjustment:** The number of volumetric integration steps per ray (N_s) was set to 32 for the baseline configuration, striking a balance between geometric fidelity and memory efficiency.

3. **Gradient Accumulation:** Training was performed with a batch size of one image per GPU, supplemented by gradient accumulation where necessary to approximate the optimization dynamics of larger effective batch sizes.

Parameter	**Standard Config**	**Our Config (T4)**	**Reduction**
GPU VRAM	32GB+	**16GB**	**-50%**
Patch Size (P_s)	64×64	32×32	**4x**
Ray Samples (N_s)	64	**32**	**2x**
Volumetric Load	**262k pts**	**32k pts**	**8x Lighter**

Table 2: Comparison showing how we optimized the architecture to fit within the memory constraints of a Tesla T4.

4.4 Training Protocol

All models (both pre-trained reproduction and retraining on the four datasets) were optimized using the Adam optimizer with standard StyleGAN2 hyperparameters ($\beta_1 = 0.0$, $\beta_2 = 0.99$). The learning rates for both the generator and discriminator were fixed at 2×10^{-3}.

For the **ablation experiments (conducted exclusively on the CARLA dataset)**, specific hyperparameters were overridden at runtime via Hydra's command-line configuration interface:

- **Regularization Ablation:** The R_1 gradient penalty coefficient was reduced using `+D_kwargs.r1_gamma=0.5`.
- **Augmentation Ablation:** Data augmentation was disabled by setting `+D_kwargs.augment_p=0.0`.
- **Geometric Ablations:** Sparse volumetric sampling and reduced discriminator receptive fields were tested via `+G_kwargs.synthesis_kwargs.n_samples=8` and `+D_kwargs.patch_size=8`, respectively.

All experimental runs were monitored using TensorBoard, tracking the Fréchet Inception Distance (FID) and adversarial loss curves to assess convergence behavior and training stability.

5 Experimental Results

In this section, we present a qualitative evaluation of the reproduced HyperNeRFGAN model across multiple datasets and analyze the effects of hyperparameter variations defined in our ablation studies.

5.1 Reference Evaluation: Official Pre-trained Models

To establish a qualitative benchmark for our reproduction efforts, we first evaluate the official pre-trained checkpoints provided by the original authors. These models were typically trained in high-resource environments (e.g., NVIDIA V100/A100 clusters) with higher rendering budgets ($P_s = 64, N_s = 64$), representing the optimal performance of the HyperNeRFGAN architecture.

Visual Quality: Figure 2 showcases random samples generated from the pre-trained weights across the four categories. The generated objects exhibit high-frequency details, sharp geometric edges, and minimal rendering artifacts. For instance, the ShapeNet Planes (Figure 2, bottom-right) display distinct engine turbines and stabilizers, details that are often smoothed out in lower-resolution training regimes.

CARLA

ShapeNet Cars

ShapeNet Chairs

ShapeNet Planes

Figure 2: Samples generated by the official pre-trained models. These results serve as a high-fidelity reference for our reproduction experiments.

Manifold Continuity (Pre-trained): We also examine the latent space continuity of these reference models. As illustrated in Figure 3, the pre-trained checkpoints produce exceptionally smooth transitions. Notably, the intermediate frames maintain strict geometric validity (e.g., a car does not lose its wheels during the transition), confirming that the hypernetwork has learned a robust mapping from $\mathcal{Z}$ space to the weight space.

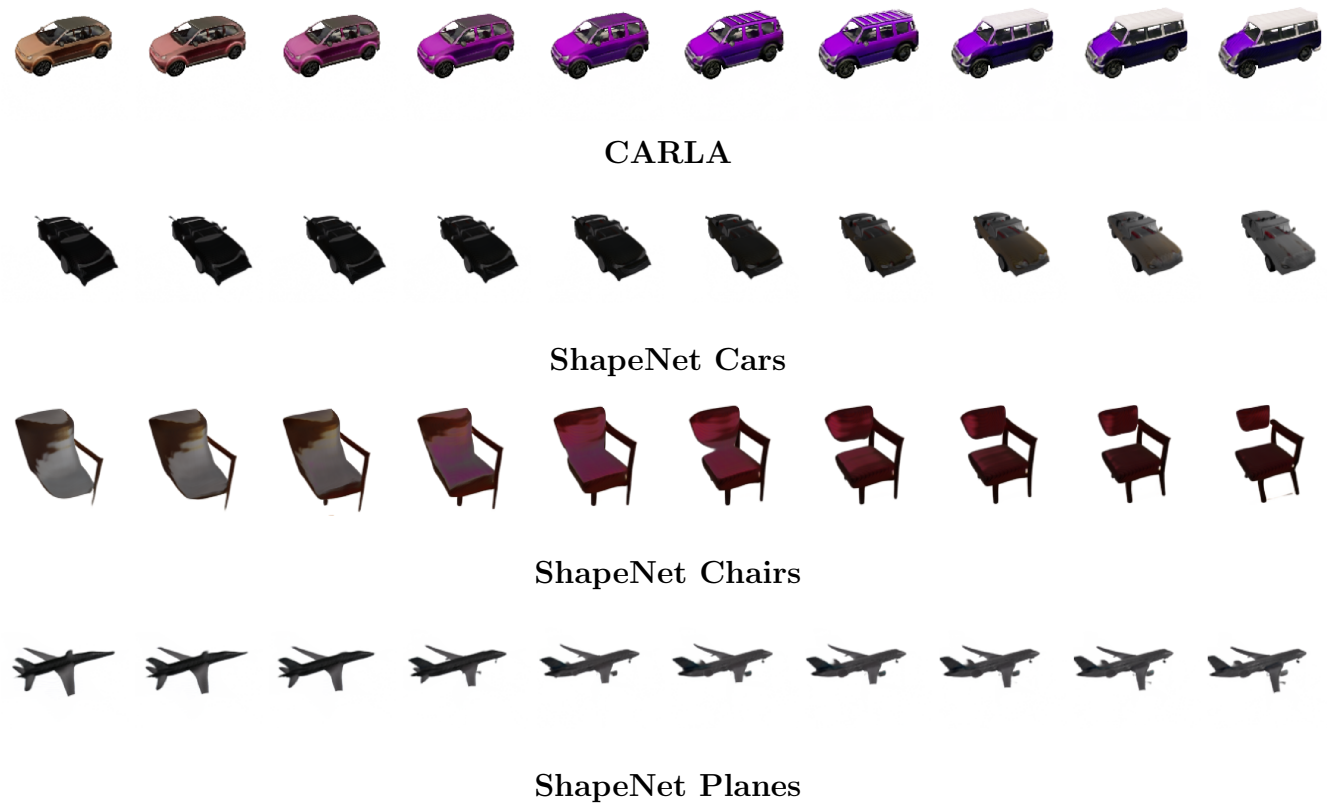


Figure 3: Latent space interpolation using official pre-trained weights. The transitions represent the theoretical upper bound of smoothness achievable by the architecture.

5.2 Reproduction Results: Low-Resource Baseline Performance across Datasets

The **baseline model**, trained with a rendering patch size of 32×32 and a ray sampling density of $N_s = 32$ for a limited duration of **200 kimg** (200,000 images), demonstrates the ability to synthesize plausible 3D geometries despite operating under this strict computational constraint. We successfully reproduced the model on **CARLA** as well as three **ShapeNet** categories (Cars, Planes, and Chairs).

- **CARLA & ShapeNet Cars:** The model captures global vehicle structure, including diverse body types and color variations.
- **ShapeNet Planes:** Thin structures such as wings and stabilizers are reconstructed reliably, though high-frequency details are smoothed due to reduced sampling.

- **ShapeNet Chairs:** Complex topologies (e.g., gaps between legs) are handled with reasonable consistency, highlighting the versatility of the view-independent NeRF representation.

CARLA **ShapeNet Cars**

ShapeNet Chairs **ShapeNet Planes**

Figure 4: Baseline qualitative results of HyperNeRFGAN across different datasets. Each image shows a randomly sampled object rendered from a fixed viewpoint.

Latent Space Interpolation: To verify that the hypernetwork learns a continuous manifold of 3D objects rather than memorizing discrete training samples, we perform linear interpolation between two latent vectors z_1 and z_2:

$$z(\alpha) = (1 - \alpha)z_1 + \alpha z_2, \quad \alpha \in [0, 1]. \tag{11}$$

The resulting image sequences (visualized primarily on CARLA) exhibit smooth semantic transitions across interpolation steps. We observe gradual changes in global geometry (e.g., transitions between sedan-like and hatchback-like body structures) as well as appearance attributes, without abrupt geometric discontinuities or "popping" artifacts. These observations indicate that the hypernetwork successfully maps the Gaussian latent space to a coherent weight space of the NeRF MLP.

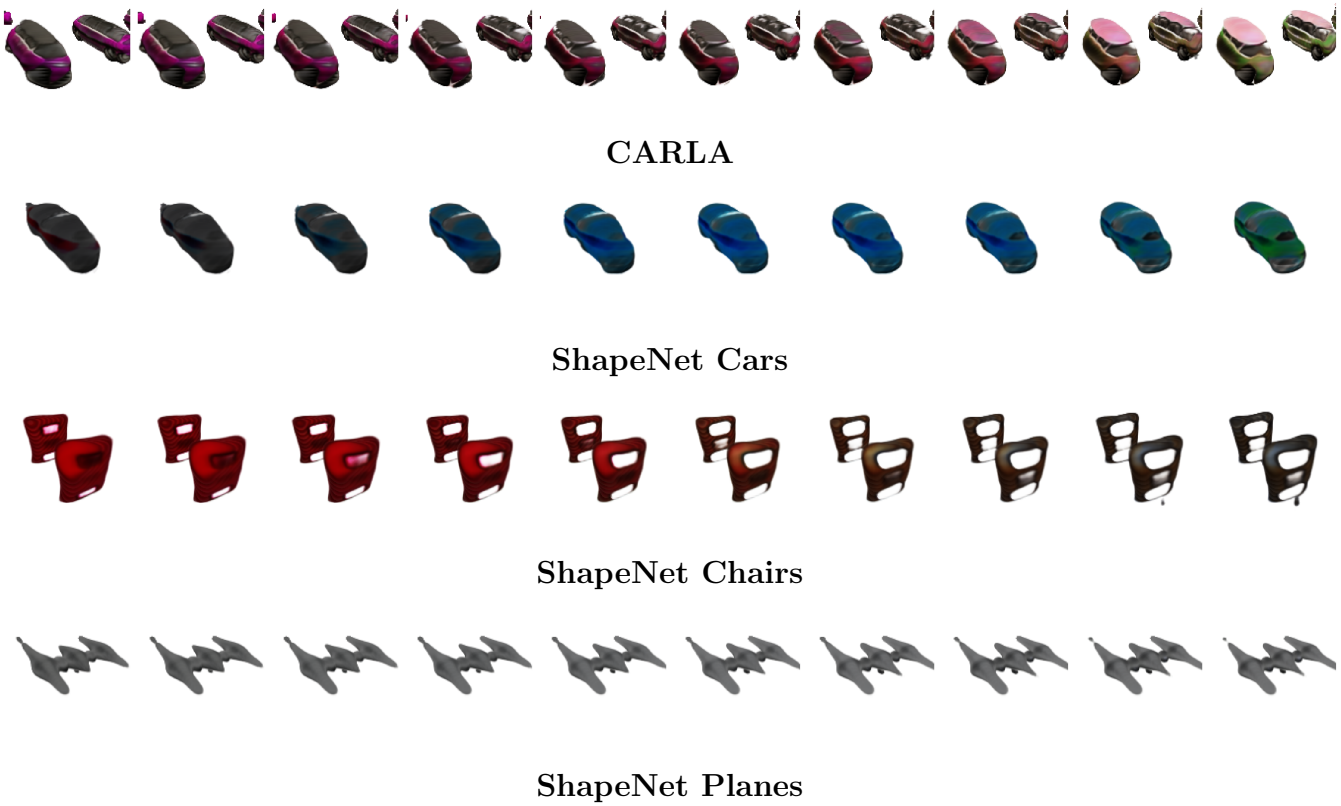


Figure 5: Linear latent space interpolation results. Each row shows interpolation between two latent codes, demonstrating smooth transitions in object identity and viewpoint.

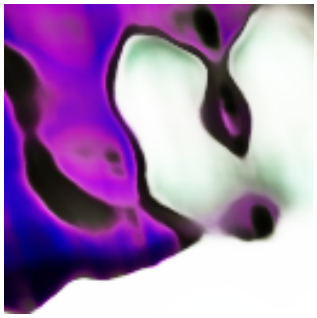
5.3 Ablation Study Analysis (CARLA Dataset)

To isolate the impact of specific hyperparameters, the following ablation studies were conducted exclusively on the **CARLA** dataset (also for a limited duration of **200 kimg** - 200,000 images). We analyze the individual impact of each modification relative to the baseline configuration.

5.3.1 Impact of Sampling Density ($N_s = 32 \rightarrow 8$)

Reducing the number of volumetric samples per ray from 32 to 8 results in a severe degradation of geometric fidelity.

(a) Baseline ($N_s = 32$)



(b) Sparse Sampling ($N_s = 8$)

Observation: Generated objects appear semi-transparent and "cloudy," lacking well-defined surfaces. Fine-grained structures such as wheels, mirrors, and sharp body contours are almost entirely absent.

Analysis: This failure mode is theoretically expected. The volumetric rendering process approximates the continuous rendering integral via numerical quadrature. With only eight samples, the spacing between consecutive integration points becomes too large to accurately capture rapid variations in the density field $\sigma(\mathbf{x})$. As a result, opacity accumulation fails to saturate, producing ghosting artifacts and weak surface definition.

5.3.2 Impact of Patch Size ($P_s = 32 \rightarrow 8$)

Reducing the rendering and discriminator patch size from 32×32 to 8×8 significantly disrupts global shape coherence.

(a) Baseline ($P_s = 32$) (b) Small Patch ($P_s = 8$)

Observation: While local textures such as metallic paint and tire-like patterns remain plausible, the global object structure becomes fragmented. Generated images often resemble disjoint collections of car parts rather than a unified 3D object.

Analysis: With a highly localized receptive field, the discriminator can only enforce local texture realism and fails to evaluate global semantic consistency. Consequently, the generator learns to optimize local appearance statistics without maintaining a physically plausible 3D configuration. This confirms that a minimum patch size is required for the discriminator to impose meaningful geometric constraints.

5.3.3 Impact of Data Augmentation ($p = 0.5 \rightarrow 0.0$)

Disabling adaptive discriminator augmentation leads to clear signs of discriminator overfitting.

(a) Baseline (Diverse)

(b) No Augment (Collapsed)

Observation: Sample diversity decreases substantially, and the generator collapses to producing a limited set of recurring car shapes, largely ignoring variations in the latent code z.

Analysis: Although the CARLA dataset is visually clean, its limited size (10,000 images) makes the discriminator prone to memorization when augmentation is disabled. As a result, the generator converges to a small number of "safe" modes that reliably fool the discriminator, rather than learning the full data distribution.

5.3.4 Impact of Regularization Strength ($\gamma = 1.0 \rightarrow 0.5$)

Reducing the R_1 gradient penalty produces mixed effects on training dynamics.

(a) Baseline ($\gamma = 1.0$)

(b) Reduced Reg. ($\gamma = 0.5$)

Observation: Training initially converges faster, but loss curves exhibit increased variance. At later stages, high-frequency noise artifacts become apparent in the generated images.

Analysis: The R_1 penalty regularizes the discriminator by encouraging smoother decision boundaries. Lowering γ allows the discriminator to form sharper, more complex boundaries, which can improve detail but also destabilize the adversarial game. In our resource-constrained setting, this instability manifests as artifacts that the generator fails to suppress.

5.4 Quantitative Analysis

Due to the limited session duration of the Colab environment, we standardized the training budget for all experiments to a fixed duration of **200 kimg** (200,000 images). While this is insufficient for full convergence (typically requiring $> 25{,}000$ kimg), it provides a controlled window to analyze early-stage convergence dynamics and relative performance.

5.4.1 Convergence across Benchmarks

We monitored the Fréchet Inception Distance (FID-1k) from initialization ($kimg = 0$) to the cutoff point ($kimg = 200$). As shown in Table 3, our reproduction achieved consistent convergence across all four datasets.

Table 3: FID-1k convergence progress across reproduced datasets (0 to 200 kimg).

Dataset	Initial FID ($kimg = 0$)	Final FID ($kimg = 200$)	Improvement (%)	Training Time
CARLA (Baseline)	339.9	155.4	**54.3%**	$\approx$ 4h 18m
ShapeNet Cars	341.3	136.2	**60.1%**	$\approx$ 5h 17m
ShapeNet Planes	294.8	182.8	38.0%	$\approx$ 5h 19m
ShapeNet Chairs	296.8	169.2	43.0%	$\approx$ 5h 14m

Analysis: The consistent reduction in FID (ranging from 38.0% to 60.1% improvement) across disparate geometries—from the thin structures of airplanes to the solid volumes of cars—confirms that the HyperNeRFGAN architecture, combined with our resource-aware optimizations ($P_s = 32, N_s = 32$), is robust and capable of learning diverse 3D distributions even on consumer-grade hardware.

5.4.2 Quantitative Impact of Ablations

To measure the severity of performance degradation caused by suboptimal hyperparameters, we compare the final FID scores of the ablation runs against the CARLA Baseline. Table 4 summarizes these findings.

Table 4: Comparison of Final FID-1k scores at 200 kimg on the CARLA dataset.

Experiment Configuration	Final FID-1k	Degradation vs. Baseline
Baseline ($N_s = 32, P_s = 32$)	**155.4**	—
Sparse Sampling ($N_s = 8$)	260.1	+104.7
Small Patch ($P_s = 8$)	324.4	+169.0
No Augmentation ($p = 0.0$)	154.9	−0.5
Reduced Reg. ($\gamma = 0.5$)	194.8	+39.4

Analysis: The quantitative results corroborate our qualitative observations:

- **Geometric Collapse:** The experiments reducing sampling ($N_s = 8$) and patch size ($P_s = 8$) resulted in significantly higher FID scores (260.1 and 324.4 respectively), mathematically confirming that the model failed to produce recognizable features suitable for Inception-v3 extraction.
- **Training Instability vs. Mode Collapse:** Interestingly, the configuration without augmentation ($p = 0.0$) achieved a final FID of 154.9, which is numerically slightly better (-0.5) than the baseline. However, as noted in the qualitative section, this metric is deceptive; it

likely reflects "mode dropping," where the generator minimizes FID by memorizing a few high-quality samples rather than learning the full distribution. Conversely, reducing regularization ($\gamma = 0.5$) yielded a **higher** FID (194.8), indicating **unstable** convergence dynamics driven by noisy gradients from the discriminator.

6 Discussion

6.1 The Efficacy of the Hypernetwork Paradigm across Domains

The successful reproduction of HyperNeRFGAN on multiple datasets—ranging from **CARLA** to diverse **ShapeNet** categories (Cars, Planes, Chairs)—validates the core premise of the original paper. It confirms that a hypernetwork can effectively serve as a generative mapping from a Gaussian latent distribution to the weight space of a 3D scene representation, regardless of the object topology.

Our latent interpolation results (Section 5.1) are particularly significant. They demonstrate that the model does not merely memorize a discrete set of 3D meshes but instead learns a continuous and navigable manifold of 3D objects. This behavior implies that the hypernetwork captures underlying semantic parameters of the dataset (e.g., shape, size, and color) without explicit supervision, a capability that standard voxel-based GANs often struggle to achieve at higher resolutions.

6.2 The Trade-off of View-Independence

A defining characteristic of HyperNeRFGAN is the omission of the viewing direction $\mathbf{d}$ from the target network F_θ.

Advantage: Our experiments confirm that this design choice simplifies convergence. By removing the need to model view-dependent effects (such as changing specular reflections), the generator can focus entirely on enforcing geometric consistency. This likely contributes to the model's robustness under a limited computational budget and explains its consistent performance across both simple (chairs) and complex (cars) geometries.

Limitation: Qualitative inspection reveals that the generated objects exhibit a predominantly matte or diffuse appearance. In particular, metallic surfaces (e.g., on CARLA and ShapeNet Cars) lack the specular highlights characteristic of real-world vehicles, such as reflections moving across curved surfaces. While this assumption is acceptable for ShapeNet Chairs or domains with largely Lambertian surfaces (e.g., medical imaging), it limits the achievable photorealism for objects with complex material properties.

6.3 The "Memory Wall" in 3D-Aware Synthesis

Our ablation studies on the CARLA dataset regarding sampling density (N_s) and patch size (P_s) expose a critical bottleneck in the broader accessibility of 3D-aware generative models, which we refer to as the *Memory Wall*. The failure of the $N_s = 8$ configuration suggests that volume rendering is fundamentally irreducible: there exists a hard mathematical lower bound on the number of samples required to approximate the volume rendering integral before the 3D illusion collapses into semi-transparent artifacts. Similarly, the failure observed with $P_s = 8$ indicates that discriminators require a minimum receptive context to enforce global geometric consistency.

This creates a practical dilemma for researchers operating with consumer-grade hardware (such as the Tesla T4 GPU used in our experiments), as increasing resolution incurs quadratic memory scaling. These findings suggest that future research should prioritize not only architectural innovations but also more memory-efficient rendering paradigms, such as *Neural Point-Based Graphics* or *Sparse Voxel Octrees*, to decouple resolution from memory cost.

6.4 Stability and Mode Coverage

The phenomena observed when relaxing regularization ($\gamma = 0.5$) or disabling data augmentation ($p = 0.0$) underscore the inherent fragility of the hypernetwork-based GAN formulation.

Specifically, the case of disabled augmentation ($p = 0.0$) presents an intriguing paradox: while it achieved a competitive FID score (Table 4), qualitative inspection revealed severe **mode collapse**. This suggests that without augmentation, the discriminator overfits to the small training set, forcing the generator to optimize for a narrow set of "safe" samples rather than learning the full distribution. Conversely, the increased FID observed with reduced regularization implies that strong inductive biases (like the R_1 penalty) are structural requirements necessary to constrain the hypernetwork to remain on the manifold of valid neural weights.

7 Conclusion

In this report, we presented a comprehensive reproduction and empirical analysis of **HyperNeRFGAN**, a generative framework that leverages hypernetworks to synthesize 3D-aware images without explicit camera pose supervision. By adapting the original architecture to a resource-constrained environment (a single NVIDIA Tesla T4 GPU), we demonstrated that the model is capable of learning continuous geometric manifolds across multiple datasets, including **CARLA** and various **ShapeNet** categories (Cars, Planes, and Chairs).

Our investigation yielded three key insights regarding the training of 3D-aware GANs under hardware constraints:

1. **The Geometry–Memory Trade-off:** Through systematic ablation studies on the CARLA dataset, we established the existence of a strict lower bound on volumetric integration quality. Reducing the ray sampling density (N_s) below 32 leads to a catastrophic degradation of surface solidity, causing generated objects to appear semi-transparent and cloud-like. Likewise, maintaining a minimum discriminator patch size ($P_s \geq 32$) is essential for preserving global structural coherence.

2. **Hypernetwork Stability:** We confirmed that generating neural network weights is an inherently sensitive optimization problem. Standard GAN stabilization techniques—in particular R_1 **regularization** and **Adaptive Discriminator Augmentation**—are not optional design choices. Weakening or removing these components ($p = 0.0$, $\gamma = 0.5$) consistently resulted in training instability and mode collapse, preventing the model from capturing the full diversity of the data.

3. **View-Independence as a Feature:** The simplified, view-independent NeRF formulation proved to be a robust and effective design choice. By eliminating the need to model view-dependent effects, the model converged reliably on unposed data across diverse object topologies, making HyperNeRFGAN a promising candidate for application domains where camera

tracking is impractical or unavailable, such as medical imaging or biological structure synthesis.

In conclusion, HyperNeRFGAN represents a meaningful advancement in implicit 3D generative modeling. While its memory requirements remain a limiting factor for high-resolution synthesis on consumer-grade hardware, our work demonstrates that, with careful optimization of the rendering budget, high-quality 3D-aware generative models can be trained without access to industrial-scale computing infrastructure. Future research should focus on integrating sparse voxel acceleration structures or alternative efficient rendering paradigms to further decouple geometric resolution from memory consumption.

References

[1] A. Kania, A. Kasymov, J. Kościukiewicz, A. Górak, M. Mazur, M. Zieba, and P. Spurek, "HyperNeRFGAN: Hypernetwork approach to 3D NeRF GAN," *arXiv preprint arXiv:2301.11631*, 2023.

[2] 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," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 405–421.

[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8110–8119.

[4] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 12104–12114.

[5] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," in *International Conference on Learning Representations (ICLR)*, 2017.

[6] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "GRAF: Generative radiance fields for 3D-aware image synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 20154–20166.

[7] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5799–5809.

[8] I. Skorokhodov, S. Ignatyev, and M. Elhoseiny, "Adversarial generation of continuous images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10753–10764.

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, 2017.