HyperNeRFGAN: Hypernetwork approach to 3D NeRF GAN

Hai-Dang Nguyen[1,2], Truong-Nguyen Dang[1,2]

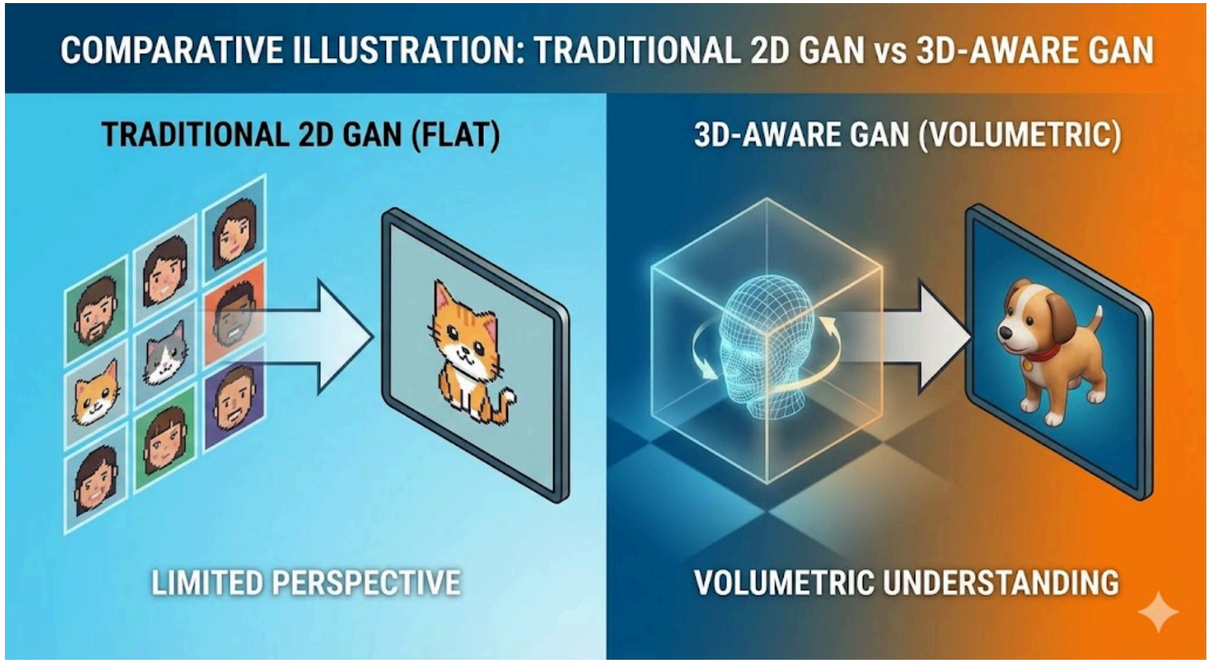
[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam.
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Ho Chi Minh City, December 28, 2025

1. INTRODUCTION

1.1 Background & Motivation

- **Context:** 3D-aware Image Synthesis combines GANs (Generative) + NeRF (3D Consistency).
- **The Problem:**
  - **Computational Cost:** Rendering requires millions of queries. Typically needs NVIDIA V100/A100.
  - **Pose Dependency:** Standard NeRF needs camera labels. Real-world data is often unposed.
- **Our Goal:** Reproduce and optimize HyperNeRFGAN on a Single Tesla T4 (16GB).



1. INTRODUCTION

1.2 Problem Formulation (Mathematical)

Input: Unstructured 2D image collection $\mathcal{D} = \{I_1, \ldots, I_N\}$. Unknown poses.

Objective: Learn a generator G and volume representation V such that:

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

- z~N(0,I): Latent code.
- ξ: Random camera pose.
- π: Differentiable Volumetric Rendering operator.

Adversarial Game:

$$\min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log(1 - D(\hat{I}))]$$

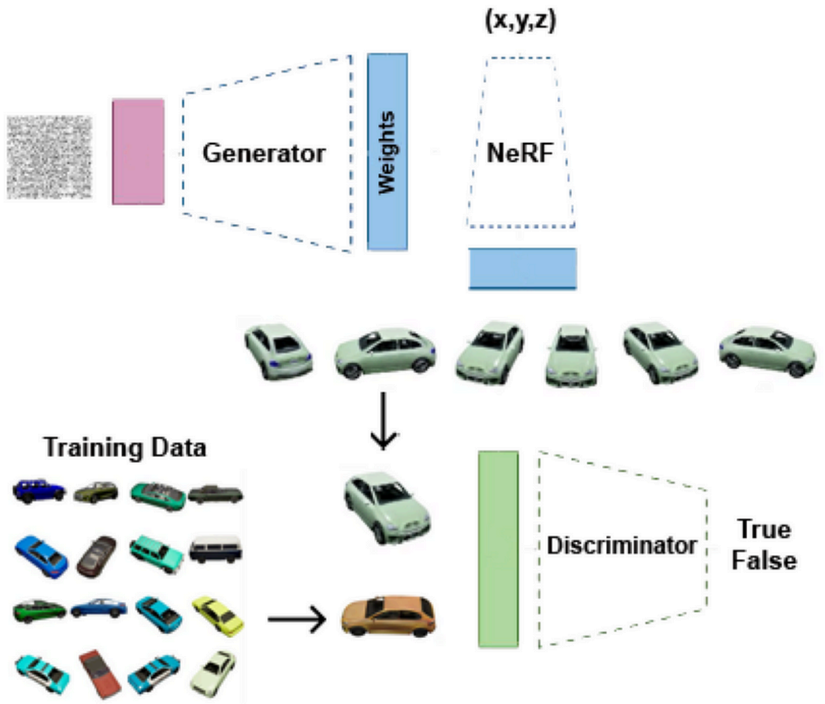
2. METHODOLOGY

2.1 Architecture Overview

1. **Hypernetwork (G):** StyleGAN2 backbone. Maps $z \rightarrow \text{Weights } \theta$.

2. **Target Network (F_θ):** Simplified NeRF MLP. Uses weights from G.

3. **Discriminator (D):** 2D CNN. Critiques rendered images.



2. METHODOLOGY

2.2 Target Network & View-Independence

1. Mathematical Formulation:

- **Standard NeRF (View-Dependent):**

$$F : (\underbrace{\mathbf{x}}_{\in\mathbb{R}^3}, \underbrace{\mathbf{d}}_{\in\mathbb{S}^2}) \longrightarrow (\underbrace{\mathbf{c}}_{\in\mathbb{R}^3}, \underbrace{\sigma}_{\in\mathbb{R}_{\geq 0}})$$

  - *Inputs:* Spatial coordinate $\mathbf{x}$ and Viewing direction $\mathbf{d}$.
  - *Outputs:* Color vector $\mathbf{c}$ (RGB) and **Scalar** Volume Density σ.

- **HyperNeRFGAN Simplification:**

$$F_\theta : \mathbf{x} \in \mathbb{R}^3 \longrightarrow (\mathbf{c}, \sigma) \in \mathbb{R}^3 \times \mathbb{R}_{\geq 0}$$

  - *Note:* The input domain is reduced from 5D ($\mathbb{R}^3 \times \mathbb{S}^2$) to 3D ($\mathbb{R}^3$).

2. METHODOLOGY

2.2 Target Network & View-Independence

2. The Lambertian Surface Assumption:

- **Definition:** Surfaces reflect light uniformly in all directions (perfectly diffuse).
- **Implication:** Emitted color is **isotropic** (independent of viewing angle).

$$\mathbf{c}(\mathbf{x}, \mathbf{d}) \approx \mathbf{c}(\mathbf{x}) \implies \mathbf{d} \text{ is redundant.}$$

3. Why this Assumption Helps?

- **Removes Ambiguity:** Without camera pose labels (unposed data), the generator struggles to distinguish between *texture patterns* and *specular reflections*.
- **Stabilizes Training:** Reduces the mapping complexity for the Hypernetwork, allowing it to focus on learning accurate **Geometry (σ)** and **Texture ($\mathbf{c}$)**.

2. METHODOLOGY

2.3 Factorized Multiplicative Modulation (FMM)

- **Challenge:** Generating full weight matrix $W \in \mathbb{R}^{n_{out} \times n_{in}}$ is $O(n^2)$. Too heavy!
- **Solution (FMM):**

$$y = W \odot (A \times B) \cdot x_{in} + b$$

- **Derivation of Efficiency:**
  - Decompose into low-rank matrices $A(n \times k)$ and $B(k \times n)$.
  - Complexity: $\mathcal{O}(k \cdot (n_{out} + n_{in}))$.
  - With $k = 10, n = 128$: Reduction factor $\approx \mathbf{6.4}\times$.

2. METHODOLOGY

2.4 Volumetric Rendering & Discretization

- **The Integral:** $C(\mathbf{r}) = \int T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t))\,dt.$
- **Riemann Sum Approximation:**

$$\hat{C}(\mathbf{r}) \approx \sum_{i=1}^{N_s} T_i(1 - e^{-\sigma_i\delta_i})\mathbf{c}_i$$

- **Mathematical Insight for Failure ($N_s = 8$):**
  - Error $\propto \delta_i$ (distance between samples).
  - Low $N_s \implies$ High $\delta_i \implies$ Opacity term $(1 - e^{-\sigma_i\delta_i})$ fails to saturate.
  - Result: **"Cloudy" Artifacts**.

2. METHODOLOGY

2.5 Objective & R1 Regularization

- **R1 Gradient Penalty:**

$$R_1(\mathcal{D}) = \frac{\gamma}{2}\mathbb{E}_x[\|\nabla_x D(x)\|^2]$$

- **Why is it necessary?**
  - Enforces **Lipschitz Continuity** on D.
  - Bounds the gradient magnitude $\rightarrow$ Prevents exploding gradients.
  - Ensures a smooth optimization landscape for the Hypernetwork.

3. Resource-Aware Optimization (Tesla T4)

- **Constraint:** 16GB VRAM (Standard NeRF-GAN needs >32GB).
- **Our Solution (The "Low-Memory Baseline"):**
  1. **Patch Size:** $64^2 \rightarrow \mathbf{32^2}$ pixels (4x memory reduction).
  2. **Sampling:** $N_s = 64 \rightarrow \mathbf{32}$ steps.
  3. **Environment:** Micromamba + Pinned Versions (Torch 1.10).

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. RESULTS & ABLATION

4.1 Pre-trained Results (4 Datasets)

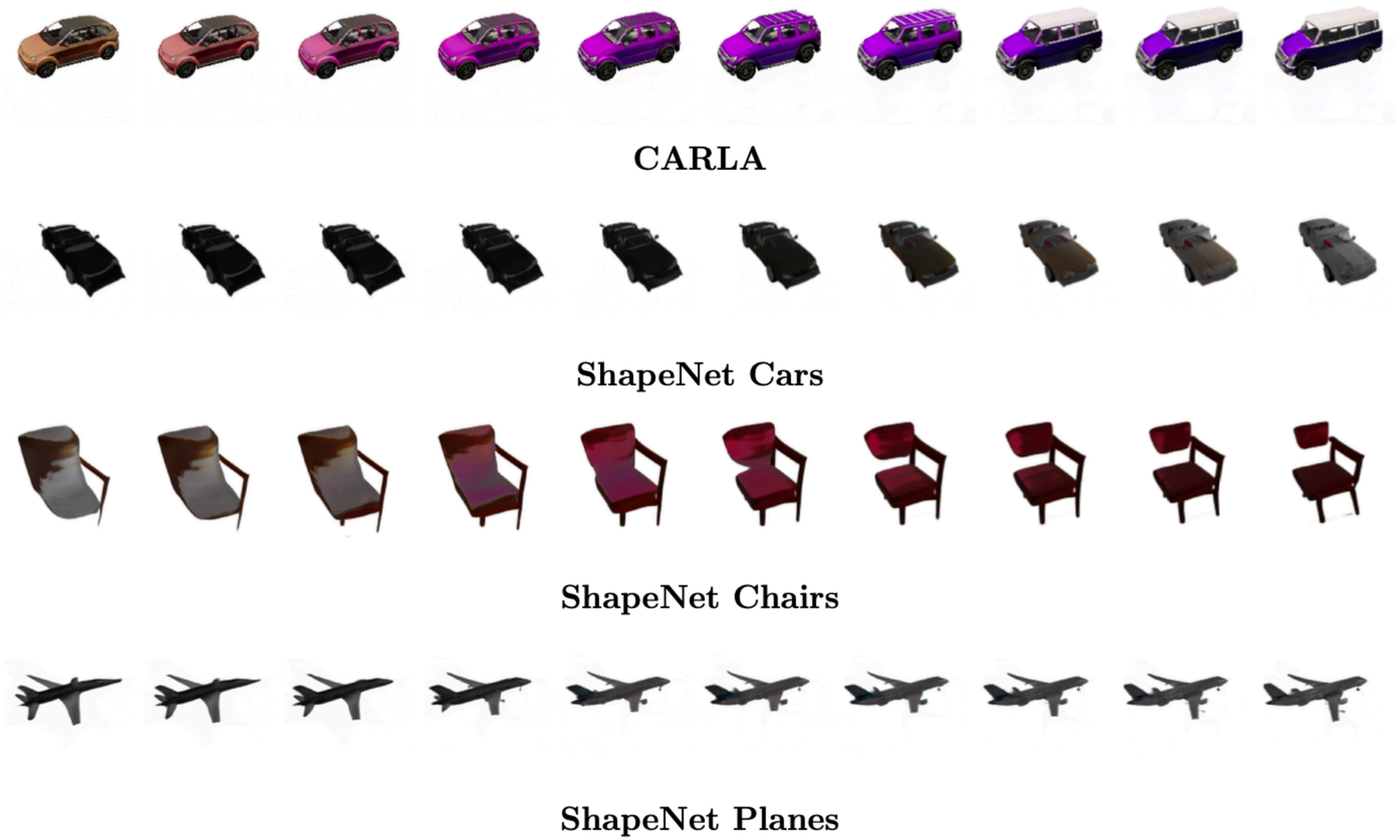


Figure 3: Latent space interpolation using official pre-trained weights. The transitions represent the theoretical upper bound of smoothness achievable by the architecture.

4. RESULTS & ABLATION

4.2 Qualitative Baseline Results (4 Datasets)

- Successfully reproduced on CARLA, ShapeNet Cars, Planes, Chairs.
- **Observations:**
  - **Cars:** Solid volumes.
  - **Planes:** Thin wings captured (Hardest task).
  - **Chairs:** Complex topology/negative space handled well.

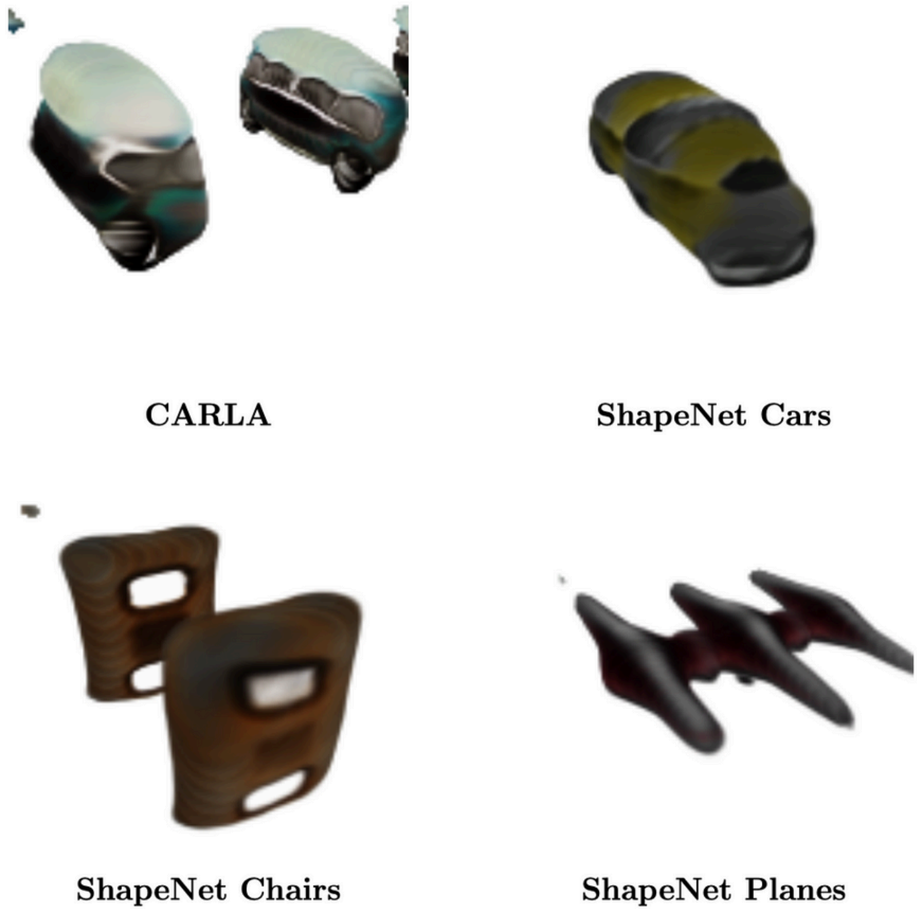


Figure 4: Baseline qualitative results of HyperNeRFGAN across different datasets. Each image shows a randomly sampled object rendered from a fixed viewpoint.

4. RESULTS & ABLATION

4.3 Manifold Continuity (Interpolation)

- Linear Interpolation: $z_{new} = (1 - \alpha)z_1 + \alpha z_2$.
- **Result:** Smooth transitions (e.g., Chair leg thickness changes gradually).
- **Proof:** The model learns the **structure**, not memorization.

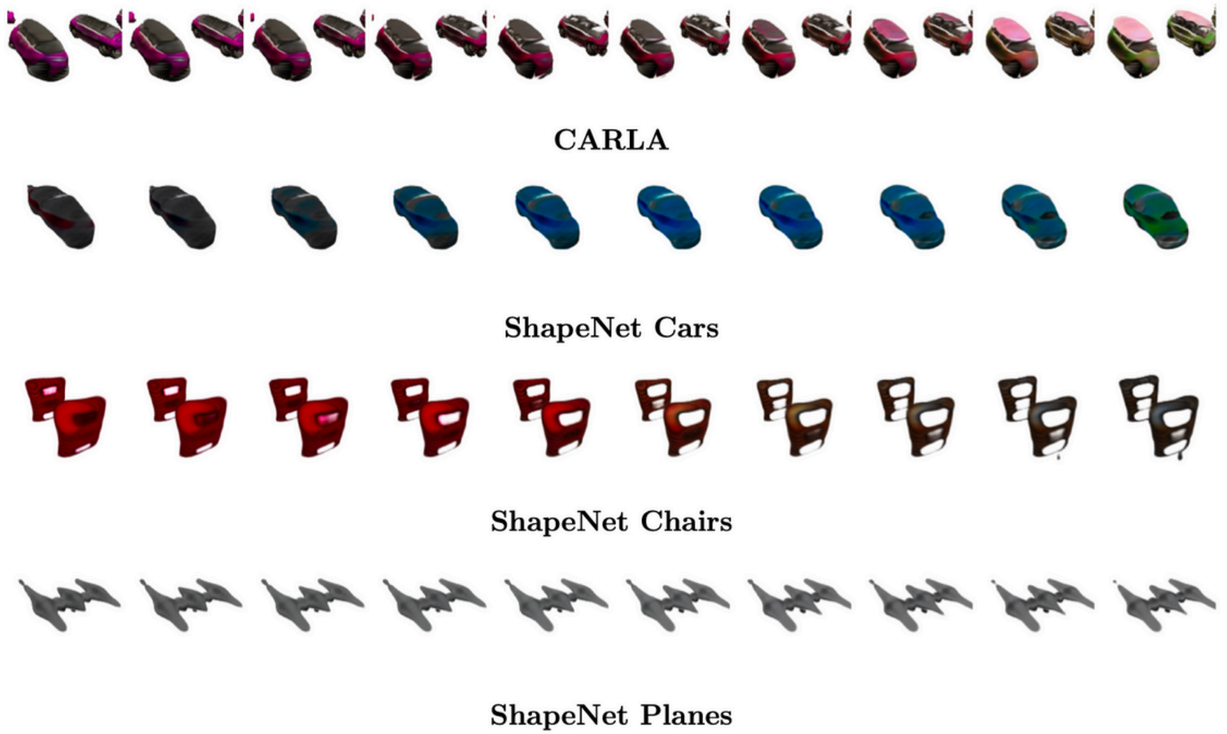


Figure 5: Linear latent space interpolation results. Each row shows interpolation between two latent codes, demonstrating smooth transitions in object identity and viewpoint.

4. RESULTS & ABLATION

4.4 Quantitative Analysis (Convergence)

Table 2: 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

- **Table:** FID improvement over 200 kimg.
- **Key Findings:**
  - **ShapeNet Cars: 60.1%** improvement (Best).
  - **ShapeNet Planes: 38.0%** improvement (Lowest - Thin structures need high N_s).
- **Baseline FID:** CARLA reached **155.4**.

4. RESULTS & ABLATION

4.5 Ablation: The Geometry-Memory Trade-off

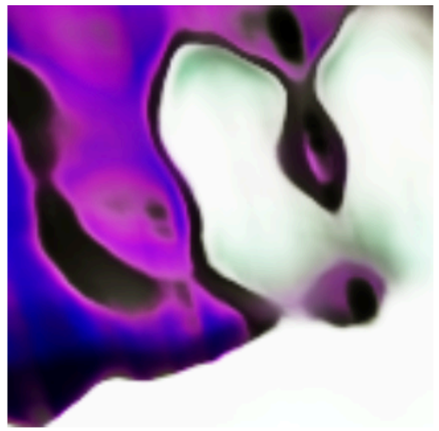
Experiment 1: Sparse Sampling (num_sample=8).

- Visual: Cloudy/Ghost-like.
- FID: **+104.7 degradation.**
- Reason: Numerical integration failure (Slide 7).

Experiment 2: Small Patch (patch_size=8).

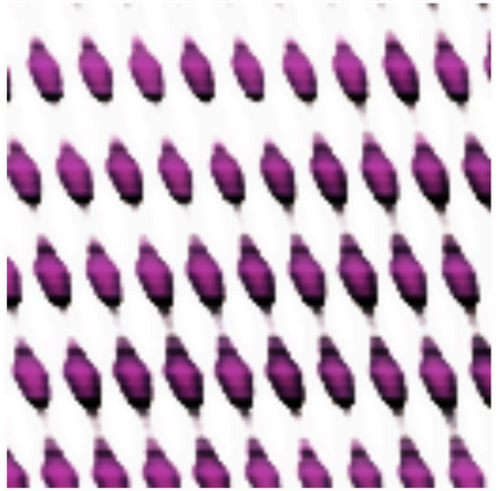
- Visual: Fragmented parts.
- FID: **+169.0 degradation.**
- Reason: D loses global context.

(a) Baseline ($N_s = 32$)



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

(a) Baseline ($P_s = 32$)



(b) Small Patch ($P_s = 8$)

4. RESULTS & ABLATION

4.6 Ablation: Stability & Mode Collapse

Experiment 3: No Augmentation (p=0.0).

- **Paradox:** FID (-0.5) is "better", BUT...
- Visual: **Mode Collapse** (All cars look the same).
- Reason: Discriminator overfitting → Generator memorizes "safe" modes.

(a) Baseline (Diverse)

(b) No Augment (Collapsed)

Experiment 4: Reduced Reg (γ=0.5).

- Visual: High-frequency artifacts.
- FID: **+39.4 degradation.**

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

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

5. CONCLUSION

- **1. The "Memory Wall":** There is a hard lower bound (num_samples >= 32) for 3D illusion.
- **2. Hypernetwork Efficacy:** Works across diverse topologies (Solid, Thin, Complex).
- **3. View-Independence:** A robust feature for unposed data, at the cost of specular reflections.
- **Final Verdict:** HyperNeRFGAN is viable on consumer hardware with careful tuning.

Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam

Thank you for your attention

Ho Chi Minh City, December 28, 2025