

# Optimal Selection for View Synthesis Tasks: NeRFs or Gaussian Splatting

Ngô Gia Lâm, Phạm Huỳnh Thiên Phú, Đỗ Trọng Hợp

Khoa Khoa học và Kỹ thuật Thông tin

Trường Đại học Công Nghệ Thông Tin, ĐHQG Tp. Hồ Chí Minh

Thành phố Hồ Chí Minh, Việt Nam

{21521054, 21521278} gm.uit.edu.vn

hopdt@uit.edu.vn

## Tóm tắt

Ngày nay, với sự phát triển của các công nghệ VR, phân cứng đồ họa máy tính,... cũng như sự gia tăng của các nhu cầu đối các thị trường liên quan tới đồ họa như giải trí, dựng hình kỹ thuật, làm phim, làm game,... đòi hỏi cần phải có những kỹ thuật xử lý đồ họa đột phá mới, có khả năng đem lại những nội dung kỹ thuật số 3d có chất lượng cao hơn. Tổng hợp khung hình mới (novel view synthesis) là một kỹ thuật quan trọng trong xây dựng đồ họa máy tính cho phép tạo ra các object 3d hoặc view 3d từ một hay nhiều bức ảnh được chụp ở nhiều góc độ khác nhau. Mảng nghiên cứu này đã gây được chú ý lớn với sự xuất hiện của NeRF cùng với những model được xây dựng dựa trên NeRF sau đó. Với đột phá của NeRF, những công nghệ VR, đồ họa máy tính, hiệu ứng CGI cùng với những nghiên cứu tương tự cũng được phát triển theo. Từ đó chúng ta có sự ra đời của Gaussian Splatting, một phương pháp novel view synthesis tốt hơn NeRF rất nhiều lần. Tuy nhiên vẫn còn một số hạn chế riêng biệt giữa cả hai phương pháp, bài báo của nhóm sẽ tập trung vào việc so sánh những phương pháp của NeRF và Gaussian Splatting để tìm ra những điểm yếu và mạnh giữa chúng, từ đó đặt chúng trong những tình huống ứng dụng phù hợp.

## 1 Giới thiệu

Bài toán Novel View Synthesis là một bài toán trong lĩnh vực Computer Vision của máy học. Bài toán này nhằm mục đích tạo ra một hình ảnh mới từ các hình ảnh đã có sẵn. Cụ thể, bài toán này yêu cầu tạo ra một hình ảnh mới từ một góc nhìn khác của cùng một đối tượng. Ứng dụng phổ biến nhất của Novel View Synthesis là lĩnh vực thiết kế đồ họa. Với kỹ thuật này, các nhà thiết kế có thể tạo ra các hình ảnh mới từ các hình ảnh đã có sẵn, giúp họ dễ dàng tạo ra các mô hình 3D và các hình ảnh động.

Việc tạo nội dung kỹ thuật số 3D tự động cho thấy nhiều ứng dụng hữu ích trong nhiều lĩnh vực

khác nhau, bao gồm trò chơi điện tử, hiệu ứng Computer-Generated Imagery (CGI), quảng cáo, phim ảnh và MetaVerse. Đồng thời cùng với sự phát triển mạnh mẽ của kỹ thuật học sau trong những năm gần đây, lĩnh vực sáng tạo nội dung 3D đã có những tiến bộ nhanh chóng.

Các nghiên cứu gần đây về tạo 3D có thể được phân thành hai loại chính: chỉ dựa trên 3d gốc (inference-only 3d native methods) và phương pháp nâng 2D thành 3D dựa trên tối ưu hóa (optimization-based 2D lifting methods)

Đối với phương pháp inference-only 3D native, tức huấn luyện trực tiếp mô hình trên bộ dữ liệu 3D. Về mặt lý thuyết, các phương pháp gốc 3D (Jun and Nichol, 2023; Nichol et al., 2022; Gupta et al., 2023) cho thấy tiềm năng tạo ra nội dung phù hợp với 3D trong vòng vài giây, mặc dù phải đánh đổi lại bằng việc phải đào tạo mô hình chuyên sâu trên bộ dữ liệu 3D vô cùng lớn. Việc tạo ra các bộ dữ liệu như vậy đòi hỏi nỗ lực to lớn của con người và ngay cả với những nỗ lực này, họ vẫn tiếp tục gặp khó khăn với những vấn đề liên quan đến như tính đa dạng và thực tế (Deitke et al., 2023; Wu et al., 2023).

Đối mặt với các khó khăn trên, nhiều nghiên cứu đã được thực hiện nhằm tìm kiếm ra các cách tiếp cận mới. Nổi bật là các nghiên cứu bằng phương pháp optimization-based 2D lifting. Có thể hiểu ta sẽ nhìn cảnh vật 3D đó từ một hoặc nhiều góc nhìn khác nhau, mỗi góc nhìn sẽ là một ảnh vật 2D. Trong đó, 2 phương pháp hiện đang cho kết quả đầy hứa hẹn là Neural Radiance Fields (NeRF) (Mildenhall et al., 2020a) và 3D Gaussian Splatting (Kerbl et al., 2023a).

Đối với phương pháp NeRF, tuy cho kết quả sinh 3D ngày càng cải thiện và triển vọng, tuy nhiên vẫn đề lớn mà phương pháp này còn gặp phải là thời gian tối ưu kém, có thể tiêu tốn hàng giờ liền để cho ra kết quả vì tốn kém của mạng NeRF, dù đã có nhiều nỗ lực để cải thiện (Müller et al., 2022; Yu et al., 2021) bằng cách tối ưu hóa Perceptron

nhiều lớp (MLP) bằng cách sử dụng tính năng dò tia thể tích để tổng hợp các cảnh được chụp ở chế độ xem mới nhưng vì việc lấy mẫu ngẫu nhiên cần thiết để hiển thị rất tốn kém và có thể gây ra nhiều nén chưa đạt được kết quả đáng kể.

Còn đối với phương pháp 3D Gaussian Splatting, là một phương pháp để thể hiện và hiển thị các cảnh không giới hạn và hoàn chỉnh với Gaussian 3D và phân tách dị hướng. Bằng cách tạo lưới (Meshes) và điểm (Points) là một trong những cách thể hiện cảnh 3D phổ biến nhất vì chúng rõ ràng và phù hợp để tạo điểm ảnh nhanh dựa trên GPU/CUDA, 3D Gaussian cho kết quả render theo thời gian thực chất lượng cao ở độ phân giải 1080p rất có triển vọng, tuy nhiên lại không đạt được chất lượng hình ảnh tốt bằng phương pháp Nerf.

Trong bài nghiên cứu này, nhóm sẽ thực hiện hai nội dung sau:

- Tìm hiểu và áp dụng hai phương pháp Neural Radiance Fields và Gaussian Splatting, thực hiện các thực nghiệm.
- So sánh, phân tích và nhận xét các kết quả thu được. Sau đó là đề xuất các hướng phát triển.

## 2 Các công trình liên quan

Đầu tiên nhóm sẽ giới thiệu tổng quan ngắn gọn về các phương pháp tái cấu trúc 3D truyền thống, sau đó trình bày những nghiên cứu trước đó về phương pháp biểu diễn Neural 3D shape.

### 2.1 Traditional Scene Reconstruction và Rendering

Các phương pháp tiếp cận novel-view synthesis đầu tiên được dựa trên trường sáng, ban đầu được lấy mẫu dày đặc (Gortler et al., 1996; Levoy and Hanrahan, 1996) sau đó cho phép việc ghi hình không cấu trúc (Buehler et al., 2001). Sự xuất hiện của Structure-from-Motion (SfM) (Snavely et al., 2006) đã mở ra một lĩnh vực mới hoàn toàn nơi mà một bộ sưu tập ảnh có thể được sử dụng để tổng hợp các góc nhìn mới. SfM ước lượng một đám mây điểm thừa thớt trong quá trình hiệu chỉnh camera, ban đầu được sử dụng cho việc trực quan hóa đơn giản không gian 3D. Tiếp theo, multi-view stereo (MVS) đã sản xuất ra các thuật toán tái tạo 3D đầy ấn tượng qua nhiều năm (Goesele et al., 2007), cho phép phát triển nhiều thuật toán tổng hợp góc nhìn (Chaurasia et al., 2013; Eisemann et al., 2008; Hedman et al., 2018; Kopanas et al., 2021). Tất cả các phương pháp này tái chiêu và pha trộn các hình

ảnh đầu vào vào camera góc nhìn mới, và sử dụng hình học để hướng dẫn việc tái chiêu này. Những phương pháp này đã sản xuất ra kết quả xuất sắc trong nhiều trường hợp, nhưng thường không thể hoàn toàn tái cấu trúc ở các khu vực chưa được tái tạo, hoặc "quá tái tạo" ("over-reconstruction"), khi MVS tạo ra hình học không tồn tại. Các thuật toán neural rendering gần đây (Tewari et al., 2022) giảm thiểu đáng kể các khuyết tật như vậy và tránh chi phí khổng lồ của việc lưu trữ tất cả các hình ảnh đầu vào trên GPU, vượt trội hơn hẳn các phương pháp này trên hầu hết các mặt trận.

### 2.2 Neural 3D shape representations

Các kỹ thuật học sâu đã được áp dụng novel-view (Flynn et al., 2015; Zhou et al., 2017); các mạng CNN cũng đã được sử dụng để ước lượng blending weights (Hedman et al., 2018), hoặc cho các giải pháp trong không gian kết cấu (Riegler and Koltun, 2020; Thies et al., 2019). Việc sử dụng hình học dựa trên MVS là một nhược điểm lớn của hầu hết các phương pháp này; thêm vào đó, việc sử dụng CNNs cho rendering cuối cùng thường xuyên dẫn đến hiện tượng "flickering" theo thời gian.

Các biểu diễn thể tích cho novel-view synthesis đã được khởi xướng bởi Soft3D (Penner and Zhang, 2017); sau đó, các kỹ thuật học sâu kết hợp với ray-marching thể tích đã được đề xuất (Hedman et al., 2018; Sitzmann et al., 2020), xây dựng trên một continuous differentiable density field để biểu diễn hình học. Rendering sử dụng ray-marching thể tích có chi phí đáng kể do số lượng mẫu lớn cần để truy vấn thể tích. Mạng neural sử dụng trường bức xạ (NeRFs) (Mildenhall et al., 2020a) đã giới thiệu kỹ thuật lấy importance sampling và positional encoding để cải thiện chất lượng, nhưng vì sử dụng một mạng Multi-Layer Perceptron quá lớn vẫn ảnh hưởng tiêu cực đến tốc độ. Sự thành công của NeRF đã dẫn đến một loạt các phương pháp tiếp theo nhằm giải quyết chất lượng và tốc độ, thường bằng cách giới thiệu các chiến lược điều chỉnh; mô hình state-of-the-art hiện nay về chất lượng hình ảnh cho tổng hợp góc nhìn mới là Mip-NeRF360 (Barron et al., 2022). Mặc dù chất lượng rendering rất nổi bật, tuy nhiên thời gian huấn luyện và rendering vẫn còn cao.

## 3 NeRF: Neural Radiance Fields

Nhóm lựa chọn hai phương pháp nổi tiếng cho tác vụ novel view synthesis là neural radiance field (Mildenhall et al., 2020b) và Gaussian Splatting



Hình 1: Những ví dụ về khả năng render của NeRF với tập dữ liệu NeRF dataset của chính tác giả cung cấp, đây là tập dữ liệu nổi tiếng thường được sử dụng trong tác vụ Novel View Synthesis

(Kerbl et al., 2023b). Sau đó nhóm tiến hành huấn luyện thử để render cho một vài sample, so sánh trên metric PSNR và khả năng render bằng fps cùng với dung lượng lưu trữ checkpoint. Từ kết quả nghiên cứu, đưa ra một vài điểm mạnh và yếu của từng mô hình, những hướng đi tiếp theo cũng như những tình huống ứng dụng có thể áp dụng trong tương lai đối với từng mô hình.

### 3.1 NeRF và các biến thể

NeRF là một kiến trúc mô hình học sâu (neural network) được thiết kế để biểu diễn không gian 3D cụ thể là các cảnh (views) dưới dạng trường bức xạ (radiance) bao gồm khả năng xác định mức sáng và màu sắc tại mọi điểm trong không gian

Mục tiêu của NeRF là xây dựng một mô hình chính xác với khả năng mô tả chi tiết và phức tạp những khung cảnh 3D sử dụng những hình ảnh không đầy đủ mọi góc nhìn. Sau khi sử dụng NeRF, mô hình sẽ giúp tổng hợp lại tất cả mọi góc nhìn trong không gian.

NeRF (Mildenhall et al., 2020b) yêu cầu một lượng lớn dữ liệu đầu vào để huấn luyện và inference. Dữ liệu huấn luyện NeRF sẽ bao gồm những hình ảnh kèm với thông tin đồ họa trong không gian, như vị trí và hướng máy ảnh khi chụp.

Điều đặc biệt ở NeRF, quá trình train diễn ra giống như render, nó sẽ cố gắng để overfit với dữ liệu được đưa vào, từ đó đưa ra toàn bộ góc nhìn có thể. Điều này có nghĩa, mỗi lần muốn inference trên một model, NeRF phải được train liên tục trên chính dữ liệu đó. Quá trình này của NeRF sẽ bao gồm tính toán các radiance (mức sáng) và màu sắc tại mọi điểm trong không gian 3D.

NeRF đặc biệt với khả năng mô phỏng các cảnh 3D (view synthesis) với độ chi tiết cao và linh hoạt trong việc tổng hợp hình ảnh tự nhiều góc



Hình 2: khả năng của NeRF in the Wild, mô hình này cho phép dựng cảnh 3D tốt hơn NeRF khi ở ngoài trời (nơi có nhiều trường bức xạ (radiance fields)). Ứng dụng của mô hình này cho phép tạo ra những hiệu ứng CGI đối với các công trình (ban đêm, ban ngày, cháy, nổ,...) trong các bộ phim

nhin khác nhau. Trên thực tế, người ta ngày càng phát hiện ra nhiều công dụng mới của NeRF ngoài mô phỏng không gian 3D. Có thể kể đến như khả năng zoom chi tiết cực cao mà lượng dữ liệu lưu trữ không cần phải quá nhiều (vài mb), khả năng tạo ra những ảo ảnh quang học trong nghệ thuật, khả năng tạo hình 3D chỉ cần 1 bức ảnh phía trước, khả năng dự đoán hình dạng vật thể,...

Sau nghiên cứu của NeRF, đã có vô số những nghiên cứu khác liên quan, góp phần thúc đẩy các ngành liên quan tới đồ họa, VR cũng phát triển. Có thể kể đến như NeRF in the wild (Martin-Brualla et al., 2020) trong dựng ảnh ngoài trời, instant NGP (Müller et al., 2022) với khả năng render cực nhanh với PSNR cao nhưng cũng có một vài nhược điểm, NeRF++(Zhang et al., 2020) (bản cải tiến hơn của NeRF về performance và tốc độ,...). Những nghiên cứu này ngoài đóng góp cho việc phát triển NeRF cũng mở đường cho những nghiên cứu mới hơn trong tác vụ Novel View Synthesis, từ đó dẫn đến sự ra đời của Gaussian Splatting, bùng nổ cho những nghiên cứu tiếp theo về tác vụ này vào cuối năm 2023.

Trong bài nghiên cứu này, nhóm sẽ tiến hành chạy mô hình trên một vài sample và render chúng, nhận xét lỗi của mô hình trong một vài trường hợp. Từ đó, nhóm rút ra ưu điểm và nhược điểm của NeRF trong thực tế khi so với Gaussian Splatting (Kerbl et al., 2023b), một mô hình mới hơn, được phát triển vào cuối năm 2023.

### 3.2 Cơ chế train của NeRF

Như đã kể ở trên, khác với các mô hình học sâu khác, train xong sử dụng một cách tổng quát cho nhiều bài toán trên nhiều ngữ cảnh, thì NeRF lại là

train tới khi overfit rồi kết thúc. Điều này xảy ra là do mục tiêu của tác vụ Novel View Synthesis là hoàn thành render cho một đối tượng, khác với các mô hình khác liên quan tới bài toán tổng quát.

Quá trình đặc biệt sẽ đòi hỏi trong mỗi trường hợp render, cần có dữ liệu cực kì tốt về một đối tượng. Nếu chất lượng độ phân giải ảnh kém, khả năng chi tiết hoá shader và 3D của mô hình sẽ cực tệ, ngược lại thì sẽ giúp khả năng chi tiết hoá cao, giúp mô hình nắm bắt được nhiều đặc trưng trong các trường bức xạ hơn. Dữ liệu của NeRF về đối tượng cần render càng phong phú càng tốt, do đó nếu quá trình preprocess cũng cần phải tránh làm mất mát dữ liệu, đặc biệt là những dữ liệu liên quan tới màu sắc và cường độ sáng.

Kỹ thuật Sampling của NeRF cũng rất đặc biệt, mô hình này sẽ cố nắm bắt những biên cảnh và vùng có sự thay đổi bất thường về ánh sáng và màu sắc trong mỗi ảnh để tạo độ chi tiết khi thay đổi góc nhìn. Từ đó, NeRF có thể đưa được ra những góc nhìn vô cùng thật, với dữ liệu đủ tốt thậm chí rất dễ đánh lừa mắt người.

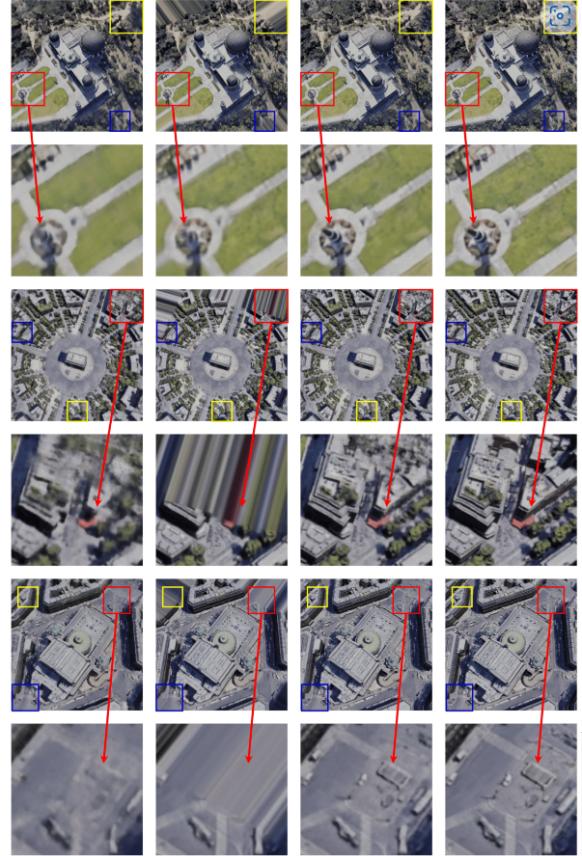
Với kỹ thuật sampling đặc biệt trên cũng tạo cho NeRF một khả năng khác ngoài chỉ là Novel View Synthesis, đó là khả năng giữ lại độ chi tiết cực cao mà dung lượng trữ ít. Một vài mô hình của NeRF sau nghiên cứu đã cho thấy khả năng vô cùng đặc biệt này, hình 3 cho thấy một vài ví dụ của khả năng này

### 3.3 Biểu diễn trường bức xạ sử dụng Neural Network

NeRF biểu diễn một cảnh bằng một hàm số có đầu vào là một vector 5 chiều, bao gồm 3 chiều vị trí trong không gian  $\mathbf{x} = (x, y, z)$  và 2 chiều chứa dữ liệu góc hướng góc nhìn  $(\theta, \phi)$  đầu ra sẽ bao gồm màu sắc  $\mathbf{c} = (r, g, b)$  and mật độ thể tích  $\sigma$ . Trong thực tế, người ta sẽ biểu diễn trên hệ đề các 3D như là một vector 3 chiều  $\mathbf{d}$ . Sau đó chúng ta tiến hành xấp xỉ vector 5D đại diện cho góc nhìn trong không gian bằng một mạng MLP  $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$  và liên tục tối ưu lại trọng số  $\Theta$  để để map lại mỗi input 5D tương ứng với mật độ thể tích và màu sắc chính xác tương ứng với mỗi góc nhìn, pipeline sẽ được biểu diễn trong hình 4 sau đây.

### 3.4 Kỹ thuật volume rendering

NeRF sẽ biểu diễn một cảnh bằng mật độ thể tích  $\sigma$  và bức xạ ánh sáng hướng ra tại mọi điểm trong không gian. Sau đó sẽ render màu sắc của các tia



Hình 3: Ví dụ về khả năng zooming trên các biến thể khác nhau của NeRF, từ trái sang phải là các nghiên cứu ngày càng mới hơn

sáng truyền qua trong cảnh bằng cách sử dụng Volume Rendering (Kajiya and Herzen, 1984). Tích phân của mật độ thể tích  $\sigma(\mathbf{x})$  sẽ được sử dụng như là xác suất của một tia sáng va chạm với một hạt cực nhỏ trong không gian tại vị trí  $\mathbf{x}$ . Màu sắc của tia sáng đó  $C(\mathbf{r})$  từ máy ảnh  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$  sẽ được tính sử dụng biên gần và xa  $t_n$  và  $t_f$  như sau:

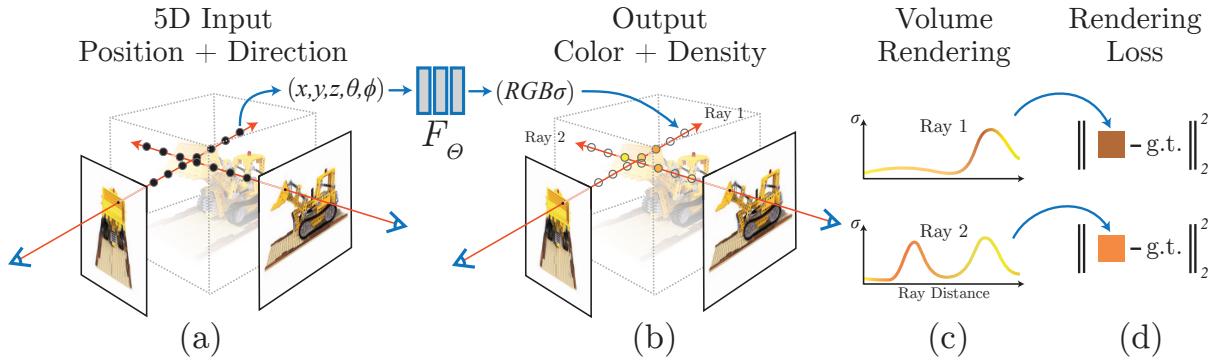
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt, \quad (1)$$

với  $T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s)) ds\right)$ .

Kỹ thuật volumne rendering rất quan trọng trong NeRF. Nó sử dụng các tia sáng mẫu, tính toán mức sáng trên tia sử dụng một biến thể công thức transmittance của định luật Beer-Lambert  $T(t)$ , sau đó tổng hợp lại màu sắc dự kiến qua  $C(\mathbf{r})$  để thu được những pixel cuối cùng của ảnh tổng hợp.

### 3.5 Optimization NeRF

Những bước trước đó nhóm đã đề cập chỉ mới giới thiệu đến những yếu tố cơ bản để xây dựng



Hình 4: Pipeline của quá trình biểu diễn góc nhìn và quy trình render. Đầu tiên ta tổng hợp ảnh bằng cách lấy mẫu toạ độ vector 5D (vị trí và góc nhìn) cũng với tia sáng máy ảnh (a), cho những vị trí này vào một mạng MLP để tạo ra màu sắc và mật độ thể tích (b), sau đó sử dụng volume rendering để kết hợp các giá trị trên thành một hình ảnh (c). Hàm rendering này có khả năng lấy đạo hàm, do đó có thể tối ưu hoá khả năng biểu diễn không giang bằng cách giảm thiểu sự chênh lệch giữa hình ảnh tổng hợp và hình ảnh thực tế (d).

một trường bức xạ sử dụng mạng neural. Tuy nhiên những bước trước đó vẫn chưa đủ để giúp mô hình xây dựng được những góc nhìn mới với hiệu suất vượt trội. NeRF sử dụng hai cơ chế mới hơn các mô hình tiền nhiệm trong tác vụ Novel View Synthesis để biểu diễn, đó là Positional Encoding và Hierarchical volume sampling. Hai cơ chế này là quan trọng trong NeRF để cải thiện khả năng biểu diễn các cảnh phức tạp với chất lượng cao hơn

### 3.5.1 Positional Encoding trong NeRF

Khác với Positional Encoding trong self-attention của Transformers với mục tiêu để giúp mô hình hiểu rõ thông tin về thứ tự của các từ hoặc đối tượng trong chuỗi đầu vào. Thì cơ chế Positional Encoding trong NeRF được sử dụng để đưa input 5 chiều tới một chiều dữ liệu cao hơn để đảm bảo nắm bắt được những điểm khác nhau trong không gian tốt hơn.

Điểm yếu của dữ liệu ít chiều được chỉ ra trong (Rahaman et al., 2018), khi mà các mạng neural có xu hướng bias cao hơn cho việc học các hàm có tần số thấp. Nghiên cứu cho rằng việc map đầu vào sang dữ liệu nhiều chiều hơn sử dụng các hàm tần số cao có thể giúp fit dữ liệu chứa những biến động tần số cao (liên quan tới chi tiết các biến thể màu sắc và thể tích không gian) tốt hơn.

NeRF biểu diễn lại  $F_\Theta$  thành tập hợp của hai hàm  $F_\Theta = F'_\Theta \circ \gamma$ , một hàm học và một không học, hiệu suất ngay lập tức được cải thiện. Khi đó,  $\gamma$  được sử dụng để map dữ liệu từ chiều  $\mathbb{R}$  tới chiều dữ liệu cao hơn  $\mathbb{R}^{2L}$ , và  $F'_\Theta$  vẫn được giữ nguyên như là một mạng MLP bình thường. Hàm encoding như sau:

$$\gamma(p) = (\sin(2^0 \pi p), \cos(2^0 \pi p), \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p)). \quad (2)$$

hàm  $\gamma(\cdot)$  sẽ được sử dụng cho 3 giá trị toạ độ về vị trí  $x$  và 2 giá trị toạ độ của vector  $d$  liên quan tới góc nhìn trong hệ trục descartes.

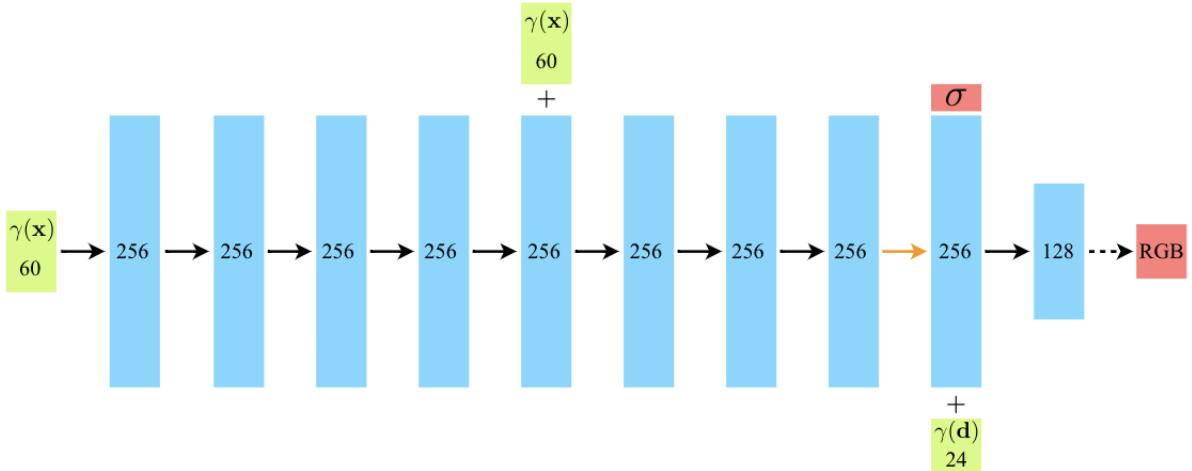
Trong những nghiên cứu sau NeRF, các biến thể thường tập trung vào việc thay đổi positional encoder. Tuỳ vào tác vụ mà chúng ta có thể tinh chỉnh positional encoder, khi màu sắc ít bị ảnh hưởng bởi ánh sáng (tranh vẽ) thì sẽ giảm số layer của  $\gamma$  hoặc tạo một hàm positional encoder đơn giản hơn. Ngoài ra việc thay đổi positional encoder cũng liên quan tới khả năng ứng dụng thực tế (InstantNGP sử dụng hashing encoder để giảm chi phí tính toán nhưng vẫn giữ được phần nào khả năng ít bị bias bởi các lầm tần suất thấp, từ đó giảm ảnh hưởng xấu tới khả năng output ánh sáng và volume density).

### 3.5.2 Hierarchical volume sampling

Việc sampling trên mỗi tia sáng trong không gian đối với từng góc nhìn sẽ khá mất thời gian và có thể kém chính xác về đầu ra. NeRF lấy cảm hứng từ một bài nghiên cứu liên quan tới phương pháp volume rendering (Levoy, 1990) và đề xuất một phương pháp tăng cường hiệu suất render sử dụng việc phân bổ các sample tỉ lệ thuận với tác động của chúng đến hình ảnh cuối cùng.

Thay vì chỉ sử dụng 1 mạng neural, NeRF sẽ liên tục optimize 2 mạng neural "coarse" và "fine" với mục tiêu là cải thiện hiệu suất mô hình trong việc biểu diễn và tạo cảnh 3D.

Mạng thô (coarse)  $\hat{C}_c(r)$  sẽ tạo những biểu diễn thô của cảnh. Nó được thiết kế để nhanh chóng ánh



Hình 5: cấu trúc mạng neural radiance fields

xạ từ không gian 3D đến một biểu diễn thô, từ đó xác định những đặc trưng lớn và độ phân giải thấp của cảnh. Về cơ bản, mạng coarse sẽ giúp đánh giá xem vị trí cần sampling nhiều nhất, từ đó tiết kiệm được thời gian thay vì phải sampling trên toàn bộ tia sáng từ máy ảnh. Những vị trí này sẽ được chọn để đưa vào mạng tinh  $\hat{C}_f(\mathbf{r})$

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i, \quad w_i = T_i(1 - \exp(-\sigma_i \delta_i)). \quad (3)$$

Ngược lại, mạng tinh (fine)  $\hat{C}_f(\mathbf{r})$  hoạt động cùng lúc với mạng thô, sẽ chú trọng vào việc tạo những biểu diễn tinh chính xác và chi tiết của cảnh. Mục tiêu sẽ là tối ưu hoá khả năng xác định các chi tiết tinh tế và độ phân giải cao hơn của cảnh. Được tính toán sử dụng công thức 1.

Việc hai mạng này liên tục được tối ưu hoá đồng thời với cùng một hàm mất mát. Hàm mất mát này sẽ bao gồm cả phần đánh giá chất lượng của mô hình trên tập dữ liệu huấn luyện và các thành phần khác nhau như hàm mất mát sai số tái tạo, hàm mất mát kiểm soát độ chi tiết và những thành phần khác để đảm bảo hai mạng đều được huấn luyện song song hiệu quả. Hàm mất mát này được viết như sau:

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[ \left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right] \quad (4)$$

Kết quả đầu ra sẽ được tạo ra bằng việc kết hợp đồng thời đầu ra của cả hai mạng neural trên.

Mạng thô nhanh chóng xác định đặc trưng lớn và mạng tinh tập trung cải thiện độ chi tiết và độ phân giải. Kết quả tổ hợp trên sẽ là một biểu diễn hoàn chỉnh và chi tiết của cảnh sẽ được render thông qua  $\hat{C}_f(\mathbf{r})$ . Ảnh 5 miêu tả cấu trúc mạng neural được sử dụng trong NeRF.

### 3.6 InstantNGP

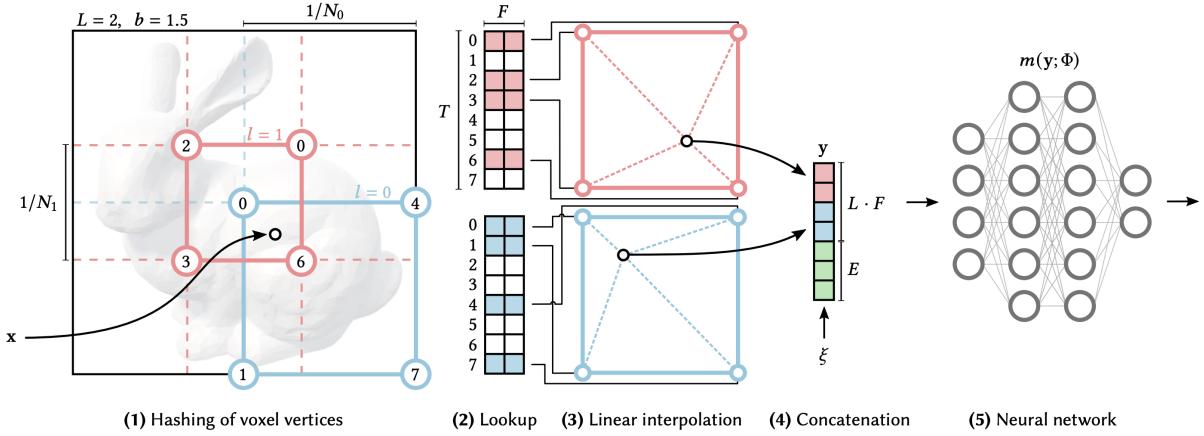
Ra đời sau NeRF, InstantNGP như một phiên bản hoàn thiện hơn rất nhiều về mặt hiệu suất. Kết hợp những kỹ thuật mới như hash encoding đa độ phân giải cho các điểm dữ liệu đầu vào, sử dụng những mô hình machine learning để giảm số lượng phép tính cần thiết để tạo ra một mô hình NeRF, InstantNGP như là một biện pháp để ứng dụng NeRF trong các tác vụ render thực tế.

Nhóm tiến hành sử dụng InstantNGP để so sánh với tiền nhiệm là NeRF, kiểm tra xem liệu hiệu suất của mô hình này có bị ảnh hưởng nhiều trong quá trình cố gắng giảm thiểu những vấn đề về khả năng render tồn đọng trong NeRF.

Do việc cài đặt của InstantNGP đòi hỏi nhiều yếu tố kỹ thuật cao (kỹ thuật lập trình CUDA) sẽ cần trở những khó khăn trong quá trình implement mô hình và debug. Tuy đã có những implement bằng PyTorch cho InstantNGP, nhưng hiệu suất chưa đạt được như sử dụng CUDA của Nvidia, vì vậy nhóm sẽ sử dụng implement của chính Nvidia trên GitHub, sau đó tiến hành so sánh hiệu suất so với NeRF và Gaussian Splatting do nhóm tự cài đặt.

## 4 3D Gaussian Splatting

3D Gaussian Splatting là một phương pháp mô hình hóa và rendering cảnh 3D dựa trên việc sử



Hình 6: Pipeline qua trình hashing cơ bản của InstantNGP.

dụng các Gaussian 3D làm primitive. Tuy nhiên, việc rendering không yêu cầu bất kỳ quá trình xử lý nặng nào - do đó có thể rendering nhanh chóng thông qua quá trình tile-based rasterizer và thu được kết quả. Kiến trúc của quá trình 3D Gaussian Plaitting được thể hiện ở hình 7

#### 4.1 Giới thiệu cơ bản về Gaussian Splatting

Trong phương pháp 3D Gaussian Splatting, bước đầu tiên là sử dụng phương pháp Structure from Motion (SfM) để ước tính đám mây điểm từ một tập hợp hình ảnh. Đây là phương pháp ước tính đám mây điểm 3D từ tập hợp hình ảnh 2D. Sau đó, chuyển mỗi điểm sẽ được biểu diễn bởi một Gaussian 3D, với các tham số bao gồm vị trí (trong không gian XYZ), ma trận hiệp phương sai (thể hiện nó được kéo dài/thu nhỏ như thế nào), màu sắc (RGB) và độ mờ (opacity  $\alpha$ ). Như vậy đã đủ để rasterization. Tuy nhiên, chỉ có thể suy ra vị trí và màu sắc từ dữ liệu SfM. Để biểu diễn những gaussian đó thành kết quả đúng với mong muốn, ta cần phải huấn luyện nó.

Quy trình đào tạo sử dụng Stochastic gradient Descent (SGD), tương tự như neural network, nhưng không có các layer. Các bước huấn luyện bao gồm:

1. Rasterize các gaussian thành một hình ảnh bằng cách sử dụng differentiable gaussian rasterization (sẽ được trình bày trong phần 4.2.1).
2. Tính toán hàm Loss dựa trên sự khác biệt giữa hình ảnh rasterized và hình ảnh ground truth.
3. Điều chỉnh các tham số gaussian theo hàm Loss.
4. Điều chỉnh tự động densification và pruning. Ở bước pruning, nếu gradient quá lớn đối với

một gaussian nhất định (nghĩa là nó sai), nếu gaussian nhỏ, ta sẽ sao chép nó, nếu gaussian lớn, ta sẽ chia nó ra và nếu alpha của gaussian quá thấp, ta loại bỏ nó.

Quy trình này giúp các gaussian phù hợp hơn với các chi tiết mịn, đồng thời cắt bớt các gaussian không cần thiết. Ví dụ như trong hình 8, là một hình phong cảnh được biểu diễn bởi khoảng 7 tỷ gaussian.

#### 4.2 Cơ chế của Gaussian Splatting

Sau đây nhóm sẽ trình bày rõ từng phần trong các bước huấn luyện trên.

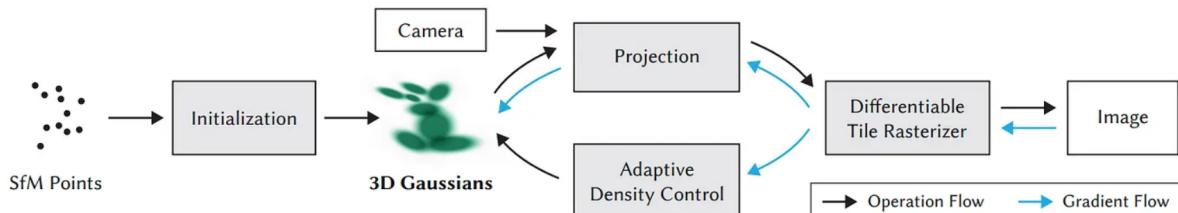
##### 4.2.1 Differentiable 3D Gaussian Splatting

Để biểu diễn cảnh vật ở chất lượng cao trong bài toán novel view synthesis bắt đầu từ một tập hợp thưa thớt (SfM) điểm không có phân phối chuẩn.(Kerbl et al., 2023a) đã kế thừa hàm primitive của differentiable volumetric representations, đồng thời unstructured và explicit để cho phép hiển thị rất nhanh. Và Gaussian 3D đã được chọn vì nó có khả năng phân biệt và có thể dễ dàng chiều thành các mảng 2D, cho phép trộn  $\alpha$ -blending nhanh để hiển thị.

Nhóm sử dụng lại mô hình hóa hình học dưới dạng một tập hợp các Gaussian 3D không yêu cầu theo phân phối chuẩn. Mỗi Gaussian được xác định bởi ma trận hiệp phương sai 3D đầy đủ  $\Sigma$  được xác định trong world space (Zwicker et al., 2001) có tâm tại điểm  $\mu$  (mean):

$$G(x) = e^{-\frac{1}{2}(x)^T \Sigma^{-1}(x)} \quad (5)$$

Gaussian này sẽ được nhân với hệ số  $\alpha$  trong quá trình blending.



Hình 7: Kiến trúc của phương pháp 3D Gaussian Platting. Bắt đầu từ các điểm thưa thớt đám mây Structure-from-Motion (SfM), quy trình tối ưu hóa đã sử dụng rendering dựa trên fast tile-based renderer và tạo ra một tập hợp Gaussian 3D, đồng thời mật độ của chúng được kiểm soát thích ứng.



Hình 8: Hình phong cảnh được biểu diễn bởi khoảng 7 tỷ gaussian.

#### 4.2.2 Optimization 3D Gaussians

Cốt lõi của phương pháp 3D Gaussian plattting là bước tối ưu hóa, sau khi tạo ra một tập hợp dày đặc các Gaussian 3D thể hiện chính xác khung cảnh cho free-view synthesis. Bằng cách tối ưu hóa các vị trí  $p$ ,  $\alpha$ , ma trận hiệp phương sai  $\Sigma$  và hệ số tương quan SH đại diện cho màu c của mỗi Gaussian để nắm bắt chính xác diện mạo view-dependent của cảnh. Việc tối ưu hóa các tham số này được thực hiện xen kẽ với các bước kiểm soát mật độ của Gaussian để có thể biểu diễn cảnh tốt hơn.

Quá trình tối ưu hóa dựa trên việc liên tục lặp lại của việc rendering và so sánh hình ảnh kết quả từ các góc nhìn huấn luyện trong bộ dữ liệu đã thu thập. Chất lượng của các tham số trong ma trận hiệp phương sai của các Gaussian 3D đóng vai trò quan trọng cho sự biểu diễn gọn gàng vì các khu vực đồng nhất lớn có thể được thu thập với một số lượng nhỏ của các Gaussian bất đồng hướng lớn.

Với hàm kích hoạt sigmoid cho  $\alpha$  và giới hạn trong phạm vi , và một hàm kích hoạt mũ cho tỉ lệ của ma trận hiệp phương sai nhằm thu được gradient mượt mà. Sau đó ước lượng ma trận hiệp phương sai ban đầu như một Gaussian đẳng hướng.

với các trục bằng với trung bình của khoảng cách đến ba điểm gần nhất. Cuối cùng sử dụng một kỹ thuật standard exponential decay scheduling tương tự như Plenoxels ([Yu et al., 2021](#)), nhưng chỉ cho vị trí. Hàm tổn thất là L1 kết hợp với D-SSIM term:

$$L = (1 - \lambda)L_1 + \lambda L_{DSSim} \quad (6)$$

Nhóm sử dụng  $\lambda = 0,2$  (theo nghiên cứu của Kerbl et al., 2023a) cho kết quả tốt nhất) trong tất cả các thử nghiệm của mình.

## 5 Thực nghiệm

Quá trình nghiên cứu của nhóm trên các model NeRFs và Gaussian Splatting diễn ra trên nhiều tập dữ liệu thực tế (LLFF, tiny-NERF, dữ liệu của instant-npg do nvidia cung cấp và dữ liệu của nhóm). Sau quá trình nghiên cứu, nhóm tiến hành benchmark sử dụng metrics là PSNR, số frame render trong một giây và dung lượng của checkpoint.

Từ kết quả nghiên cứu, nhóm sẽ tiến hành rút ra những kết luận và điểm yếu của từng mô hình trong những trường hợp nhất định, từ đó chỉ ra những ứng dụng tối ưu nhất đối với từng mô hình.

Đối với tập custom dataset do nhóm tạo, để có thể đưa ra những input từ các ảnh và video, nhóm sẽ sử dụng COLMAP structure from motion ([Schönberger and Frahm, 2016](#)) để tái tạo lại không gian. với mỗi iteration sẽ là một lần chọn ngẫu nhiên 1 batch tia sáng từ camera ánh sáng từ tập các pixels trong dataset, sau đó inference với config phù hợp tương ứng với 2 loại model thử nghiệm custom dataset là NeRF và InstantNGP. Điểm chung của config là batch size tia sáng sẽ là 4096,  $N_c = 64$ ,  $N_f = 128$ , Optimizer là Adam ([Kingma and Ba, 2014](#)) với tham số mặc định trong pytorch ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , và  $\epsilon = 10^{-7}$ ), learning rate bắt đầu từ  $5 \times 10^{-4}$  sau đó giảm dần theo cấp số mũ tới  $5 \times 10^{-5}$ . Quá trình training kết thúc khi đạt iteration = 10000.

## 5.1 Metrics

Nhóm sử dụng PSNR, số frame render trong một giây và dung lượng của checkpoint. Sau đây sẽ là một vài chú thích cho từng metrics

Peak Signal-to-Noise Ratio (PSNR) là một độ đo được sử dụng rộng rãi trong lĩnh vực xử lý hình ảnh và video để đo lường chất lượng của hình ảnh hoặc video đã được tái tạo (tức là đã được nén và sau đó giải nén). Nó đánh giá chất lượng theo từng kênh màu, điều này có một chút tương đồng với mô hình Neural Radiance Fields (NeRF) và mô hình các mô hình màu phụ thuộc vào góc nhìn. Điều quan trọng là giá trị PSNR càng cao, chất lượng của hình ảnh, video hoặc NeRF tái tạo càng tốt.

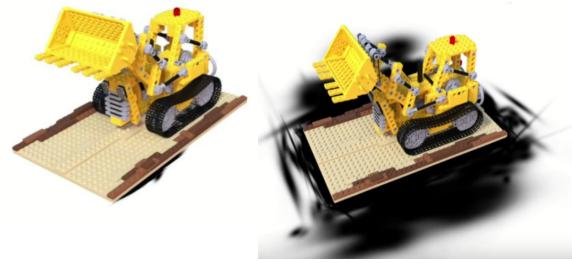
Số frame mỗi giây (fps) nhóm sử dụng sẽ là số frame để mỗi mô hình render trong một giây cùng một đoạn video resolution 800x600 60fps dài 4 giây với cùng quality với các view giống nhau trong không gian.

Dung lượng của checkpoint của mỗi mô hình cũng là một yếu tố quan trọng. Nghiên cứu (Müller et al., 2022) cho thấy khả năng thay thế của các mô hình NeRF cho việc lưu trữ những ảnh chất lượng cao với số lượng pixel khổng lồ (gigapixel images). Những mô hình NeRF sau khi train với dữ liệu là gigapixel images cho ra công dụng phóng to ảnh y hệt những dung lượng lưu trữ lại rất nhỏ.

## 5.2 Benchmark khi render NeRF synthetic

NeRF synthetic sau khi được công bố trong mảng trong view synthesis giống như "hello word" của lập trình hay imdb datset trong tác vụ Sentiment Analysis vậy. NeRF synthetic với dữ liệu là những object được công bố trong paper đầu tiên về NeRF, những object này tuy khá nhanh để render nhưng lại có những chi tiết nhỏ vô cùng khó để hoàn chỉnh những view cho đúng màu sắc và ánh sáng. Những tác giả của NeRF đã sử dụng tập dữ liệu này để làm nổi bật hơn khả năng của mô hình so với những mô hình tiền nhiệm. Gaussian Splatting là một mô hình ra đời sau NeRF được kì vọng nắm bắt tốt hơn về màu sắc cũng như chi tiết (do khả năng sử dụng các đa giác phân phối chuẩn để render), việc thử nghiệm trên tập dữ liệu này kì vọng rằng chính Gaussian Splatting sẽ đánh bại NeRF. Kết quả giữa NeRFs và Gaussian Splatting nằm trong hình 9

Kết quả khi thử nghiệm cho thấy màu sắc của Gaussian Splatting cho ra rất tốt nhưng lại có nhiều màu khi được phóng xa ra rất nhiều, tuy nhiên quá trình benchmark lại không nắm bắt điểm yếu này của Gaussian Splatting.



Hình 9: So sánh giữa NeRF và Gaussian Splatting trong việc render xe lego trong dữ liệu NeRF-synthetic. Bên trái là NeRF và bên phải là Gaussian Splatting

Tuy nhiên tập dataset này tập trung chủ yếu vào khả năng render các góc nhìn khác của object chứ chưa thực sự chú trọng đến khả năng render cảnh, thứ mà có vẻ như Gaussian Splatting vẫn thiếu. Nhóm sẽ tiếp tục sử dụng tập dataset LLFF và dataset cá nhân để thử khả năng của các model này.

## 5.3 Thủ nghiệm trên một vài dataset khác

Nhóm sử dụng tập dữ liệu LLFF để test khả năng render cảnh của NeRF, InstantNGP và Gaussian Splatting. Tập dữ liệu này bao gồm những cảnh ở ngoài môi trường, nơi có nhiều nhiễu hơn ở màu sắc thứ có thể đánh giá tốt hơn khả năng xây dựng góc nhìn từ màu sắc và ánh sáng của các model trong tác vụ Novel View Synthesis.

Nhóm đã tạo một vài custom sample để inference và đánh giá mô hình. Tập dữ liệu của nhóm bao gồm một vài object (gundam, organ, workspace) với những thông số hình ảnh khác nhau (720p, 1080p và video 720p). Còn lại là những dataset nổi tiếng khác nhiều hình ảnh, một vài dataset chỉ với một ảnh, render toàn bộ góc nhìn còn lại. Vì những cảnh thật đòi hỏi phần cứng khá cao, việc deployment trên colab và kaggle còn nhiều hạn chế do lỗi liên quan tới shell trên issue của những repository vẫn chưa được giải đáp (tháng 1/2024), nên nhóm quyết định configurate lại mô hình một chút (giảm số lượng layer của nn, scaling config và scaling trong quá trình transform ảnh để tạo thuộc tính) để tiến hành render trên máy tính local (Nvidia GTX 1650 4GB VRAM).

Ở tập custom dataset do nhóm thực hiện, do chất lượng của camera cũng như kỹ thuật chụp quay dữ liệu vẫn chưa được tốt nên kết quả khi đánh giá PSNR cũng không được trực quan. Kết quả PSNR nhóm sử dụng data gốc làm mẫu đánh giá thì cho kết quả từ 20-32 (khá tốt nhưng kết quả cũng khá phụ thuộc vào sample lựa chọn). Cụ thể là các



Hình 10: Kết quả render của NeRF và Gaussian Splatting trên 1 sample của LLFF dataset. Có thể thấy có một vài lỗi ở màu sắc những nơi màu sắc thay đổi đột ngột (màu tối sang sáng và ngược lại) khi thay đổi góc nhìn xảy ra ở Gaussian Splatting

sample liên quan tới phản xạ sáng hay trong suốt thì kết quả lại khá tệ. Khi sử dụng dữ liệu của những góc quay khác dataset (giống với các tập dữ liệu trước đó) thì kết quả cho ra dưới 10 PSNR, kết quả này rất tệ do chất lượng của dữ liệu và sẽ không có khả năng đánh giá tốt mô hình. Vì vậy, nhóm sẽ tiến hành sử dụng kết quả trung bình khi so sánh trên các góc nhìn gốc. Dưới đây là một vài kết quả đạt được của nhóm. Kết quả của nhóm khá khác với paper của các tác giả, điều này được giải thích bởi các tác giả trong issues trên github là do cấu hình của GPU và hệ điều hành. Gaussian Splatting nhóm không thể sử dụng dữ liệu cá nhân để benchmark vì do dữ liệu đầu vào không phù hợp (Gaussian Splatting đòi hỏi camera phải là model có PINHOLE). Kết quả nghiên cứu của nhóm nằm trong bảng 1

## Kết luận

Sau thực nghiệm và đánh giá, khả năng render cảnh của Gaussian Splatting gấp một vài lần để với màu sắc và cường độ sáng. Có vẻ là do khi sử dụng những phân phối chuẩn để làm màu sắc vô tình làm ảnh hưởng đến những góc nhìn khi mà màu sắc thay đổi đột ngột. Hình 10 cho thấy lỗi của Gaussian Splatting trong khi render cảnh ở dataset LLFF. Những lỗi này có thể do sự thay đổi về mặt định dạng dữ liệu và render. Gaussian Splatting tập trung vào rasterization nên chỉ hỗ trợ cho camera có lỗ kim, ngược lại thì LLFF lại là camera front forward, để render được tập dữ liệu, nhóm đã phải tinh chỉnh config và trích xuất thuộc tính lại. Có thể đây cũng ảnh hưởng ít nhiều đến hiệu suất của quá trình thực nghiệm.

Tuy vậy, khả năng render ở các object của Gaussian Splatting vẫn rất đáng nể, kết quả vẫn cho thấy sự vượt trội ở tác vụ render view tập trung ở một

object, tuy rằng nhược điểm checkpoint có dung lượng quá cao vẫn là một hạn chế. Khả năng của NeRF và biến thể InstantNGP thì lại vượt trội hơn khi render các view, các mô hình ít bị nhiễu về màu sắc hơn rất nhiều.

Nhìn chung thì cả 2 mô hình NeRF đều cho kết quả khá tương tự nhau, Gaussian Splatting thì vượt trội ở điểm PSNR, cho ra màu sắc và ánh sáng khá chính xác. Tuy nhiên vẫn đề khi thực nghiệm cho thấy thì Gaussian Splatting khi zoom ra xa sẽ có rất nhiều nhiễu, điều này có thể khắc phục bằng cách segmentate data trước khi cho vào huấn luyện. Thời gian để render là đặc biệt nhất, NeRF và InstantNGP tuy không có nhiều sự khác biệt về PSNR nhưng thời gian render của instantNGP lại cực kì nhanh, vượt qua hẳn NeRF, điều này được thể hiện qua số frame render được trong một giây. Bài báo gốc của tác giả cho thấy nhanh hơn 100 lần nhưng có thể do quá trình implement hoặc do liên quan phần cứng, kết quả của nhóm chênh lệch dù nhiều nhưng chỉ mới 4-5 lần tốc độ render. Ngược lại những mô hình càng nhanh có dung lượng checkpoint cực lớn, đặc biệt là Gaussian Splatting, dung lượng của checkpoint cực kì lớn nếu so với hai mô hình NeRF.

Kết quả cho thấy, cần phải lựa chọn những mô hình khác nhau đối với những tác vụ khác nhau. Nếu cần độ chi tiết cao ở mọi góc nhìn (ánh sáng và màu sắc) thì nên lựa chọn Gaussian Splatting, tuy nhiên cần phải sử dụng Segmentation để loại bỏ đi những nhiễu do Gaussian Splatting rất nhạy cảm với nhiễu. Nếu ưu tiên tốc độ render như trong các trường hợp ứng dụng CGI cho phim nhiều frame, VR view synthesis thì nên sử dụng Gaussian Splatting, nhiều nghiên cứu cho thấy Gaussian Splatting dễ dàng đạt trên 60fps ở các tác vụ này (nhóm có một vài trường hợp không đạt kết quả này do cấu hình máy). InstantNGP sẽ phù hợp nhất với các tác vụ render cảnh (nơi có nhiều nhiễu do không tập trung vào 1 object), lúc này instantNGP sẽ có hiệu suất gần ngang với NeRF nhưng lại có tốc độ render nhanh hơn rất nhiều. NeRF sẽ phù hợp nhất nếu người dùng muốn cảnh chi tiết nhất với nhiều nhiễu, tối ưu hóa dung lượng hoặc sử dụng để tinh chỉnh cho những nghiên cứu khác.

View Synthesis là một mảng nghiên cứu cho thấy rất nhiều ứng dụng. Những mô hình vừa rồi: NeRF ([Mildenhall et al., 2020b](#)), Instant-NGP([Müller et al., 2022](#)) và Gaussian Splatting ([Kerbl et al., 2023b](#)) là những cột mốc quan trọng trong những năm vừa qua trong tác vụ Novel View

Model	NeRF Synthetic			LLFF			Custom Data		
	PSNR	FPS	Storage	PSNR	FPS	Storage	PSNR	FPS	Storage
NeRF	30.52	12.7	12.7	24.13	4.9	19.3	26.51	2.3	22.3
InstantNGP	30.09	21.4	48.1	23.82	17.1	61.3	25.81	12.3	71.8
GaussianSplatting	33.28	73.3	628.3	23.17	42.9	1182.4	x	x	x

Bảng 1: Kết quả so sánh với các mô hình trong bài. Kết quả là độ đo trung bình khi render toàn bộ view trong từng tập dữ liệu sử dụng độ đo là PSNR, số frame trong một giây, dung lượng của checkpoint (mb). Những ô X là do yêu cầu của Gaussian Splatting phải là camera pinhole nên nhóm không thể sử dụng dữ liệu cá nhân để test được.

Synthesis. Từ cơ sở (NeRF) vượt trội hơn những nghiên cứu tiền nhiệm đến InstantNGP, phiên bản phát triển hơn của NeRF trong ứng dụng cho tới Gaussian Splatting, một nghiên cứu mới đây, đột phá về cả cơ chế lẫn khả năng ứng dụng, tác vụ Novel View Synthesis từ việc chỉ nghiên cứu đang dần cho thấy khả năng ứng dụng thực tế. Những điều này thúc đẩy phát triển của các ngành giải trí, thực tế ảo, phim ảnh, trò chơi điện tử,... ngày càng tạo nên những đột phá mới. Chỉ sau khi Gaussian Splatting phát hành 1.5 tháng đã có hàng trăm bài báo liên quan với các nhóm nghiên cứu tới từ các tập đoàn lớn như Google, Meta, Nvidia, LumaAI cho đến các đại học lớn nhỏ khắp thế giới, điều này cho thấy những tích cực trong công cuộc xây dựng những ứng dụng sử dụng View Synthesis, một tương lai nơi thế giới con người có thể xây dựng bởi những công nghệ thực tế ảo.

## Trích dẫn

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. [Mip-nerf 360: Unbounded anti-aliased neural radiance fields](#).

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proc. SIGGRAPH*.

Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153.

Martin Eisemann, Bert Decker, Marcus Magnor, Philippe Bekaert, Edilson Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. 2008. [Floating textures](#). *Computer Graphics Forum*, 27:409 – 418.

John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2015. [Deepstereo: Learning to predict new views from the world’s imagery](#).

Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven Seitz. 2007. [Multi-view stereo for community photo collections](#). pages 1–8.

Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. 1996. [The lumigraph](#). In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM.

Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 2023. [3dgen: Triplane latent diffusion for textured mesh generation](#).

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. [Deep blending for free-viewpoint image-based rendering](#). volume 37, pages 1–15.

Heewoo Jun and Alex Nichol. 2023. [Shap-e: Generating conditional 3d implicit functions](#).

James T. Kajiya and Brian P. Von Herzen. 1984. Ray tracing volume densities. *Computer Graphics (SIGGRAPH)*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023a. [3d gaussian splatting for real-time radiance field rendering](#).

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023b. [3d gaussian splatting for real-time radiance field rendering](#).

Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#).

Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. 2021. Point-based neural rendering with per-view optimization.

Marc Levoy. 1990. Efficient ray tracing of volume data. *ACM Transactions on Graphics*.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*.

- Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2020. *Nerf in the wild: Neural radiance fields for unconstrained photo collections*.
- Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortíz-Cayón, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020a. *Nerf: Representing scenes as neural radiance fields for view synthesis*.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020b. *Nerf: Representing scenes as neural radiance fields for view synthesis*.
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. *Instant neural graphics primitives with a multiresolution hash encoding*. *ACM Transactions on Graphics*, 41(4):1–15.
- Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. *Point-e: A system for generating 3d point clouds from complex prompts*.
- Eric Penner and Li Zhang. 2017. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11.
- Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Dräxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron C. Courville. 2018. On the spectral bias of neural networks. In *ICML*.
- Gernot Riegler and Vladlen Koltun. 2020. *Free view synthesis*.
- Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2020. *Scene representation networks: Continuous 3d-structure-aware neural scene representations*.
- Noah Snavely, Steven Seitz, and Richard Szeliski. 2006. *Photo tourism: exploring photo collections in 3d*. *acm trans graph* 25(3):835–846. *ACM Trans. Graph.*, 25:835–846.
- Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhoefer, and Vladislav Golyanik. 2022. *Advances in neural rendering*.
- Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. *Deferred neural rendering: Image synthesis using neural textures*.
- Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. 2023. *Omniorb3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation*.
- Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qin-hong Chen, Benjamin Recht, and Angjoo Kanazawa. 2021. *Plenoxels: Radiance fields without neural networks*.
- Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. *Nerf++: Analyzing and improving neural radiance fields*.
- Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. 2017. *View synthesis by appearance flow*.
- Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. 2001. *Ewa volume splatting*. *Visualization and Computer Graphics, IEEE Transactions on*, 8.

## A Nhữn Dataset trong bài

### A.1 NeRF Synthetic

NeRF synthetic dataset do chính tác giả của bài báo ([Mildenhall et al., 2020a](#)) cung cấp để demo NeRF. Tập dataset tập trung vào các object riêng biệt, tuy ít nhiều nhưng sẽ đánh giá tốt khả năng render toàn cảnh xung quanh object. Đây là trọng điểm của tác vụ Novel View Synthesis, tạo ra các góc nhìn mới. Tập dataset này cũng chính là tiêu chuẩn đánh giá trong nhiều paper liên quan tới tác vụ Novel View Synthesis sau này. [Ảnh 11](#) cho thấy một vài sample của dataset.



Hình 11: Một vài sample của NeRF synthetic dataset

### A.2 LLFF

LLFF dataset do chính tác giả của bài báo ([Mildenhall et al., 2019](#)) cung cấp, đây cũng chính

là tác giả của NeRF. Tập dataset tập trung vào việc đánh giá các cảnh ngoài trời hoặc trong nhà chứ không tập trung vào object. Review cảnh ngoài trời cũng chính là cũng chính là một trở ngại lớn khi có rất nhiều nhiễu liên quan tới màu sắc trong quá trình thay đổi góc nhìn. Tập dataset này sẽ giúp đánh giá tốt cách positional encoding hoạt động, đồng thời cũng xem phương pháp gaussian splatting liệu rằng có khả năng đủ nhạy để tìm ra thay đổi lớn giữa các dữ liệu đầu vào. [Ảnh 12](#) cho thấy một vài sample của dataset.



Hình 12: Một vài sample của LLFF dataset

### A.3 Custom Dataset

Custom dataset do nhóm tạo ra để so sánh những khả năng của các mô hình NeRFs và Gaussian Splatting trong các tác vụ đặc biệt, tuy nhiên do điều kiện camera liên quan tới quá trình render rasterize nên chưa thể áp dụng cho Gaussian splatting được. Các tác vụ của tập dataset này tập trung vào khả năng render phản xạ ánh sáng, khả năng render nhiều object trong 1 view, khả năng render vật trong suốt và khả năng render màn hình thiết bị điện tử. Nhìn chung thì do chất lượng của camera vẫn khá tệ, cũng như muốn tạo dataset cho NeRF cần phải có kĩ thuật chụp ảnh hay quay video tốt nên tập dataset này chỉ có thể đánh giá khách quan PSNR trên chính những view gần với view gốc. [Ảnh 13](#) cho thấy một vài sample của dataset.



Hình 13: Một vài sample của dataset của nhóm