Neural Radience Field (NeRF)

- <u>continuous neural representation</u> of a 3D scene
  - represents a <u>scene as a function parameterized by a neural network</u>

- model the scene as a radiance field
- neural network f_θ
  - input
    - 3D position $x = (x, y, z)$
    - viewing direction d
  - output
    - volume density σ
    - emitted color c
  - training objective
    - minimize photometric reconstruction error
- **Volumetric rendering**
  - pixel color: obtained by integrating color and density along a camera ray
  - encourages consistency across multiple views
- **Positional encoding**
  - $\gamma(x) = (\sin(2^0\pi x), \cos(2^0\pi x), \dots, \sin(2^{L-1}\pi x), \cos(2^{L-1}\pi x))$
  - applies high-frequency Fourier features to inputs
    - maps low-dimensional coordinates into a high-dimensional feature space
    - where linear models (or MLPs in the NTK regime) can express complex functions
      - can be analyzed through the NTK perspective
  - enables the MLP to represent fine details
    - vanilla MLPs exhibit spectral bias $\rightarrow$ prefer learning low-frequency functions
    - well-adapted to the data which has high-frequency variation
- limitations
  - slow training and rendering
  - requires many views
  - hard to control ($\therefore$ neural network)

Neural Tangent Kernel (NTK)

$$K(x, x') = \nabla_\theta f_\theta(x)^\top \nabla_\theta f_\theta(x')$$

- a theoretical framework to analyze infinitely wide neural networks
  - as network width $\to \infty$
    - training dynamics become linear in parameters
    - neural network behaves like a kernel method
  - defines a kernel induced by a neural network architecture
- training interpretation
  - gradient descent on the network = kernel regression with NTK
    - the induced kernel = Neural Tangent Kernel (NTK)
  - explains why overparameterized networks are easy to optimize
- limitations
  - does not fully capture feature learning in finite-width networks
  - more descriptive than predictive for modern deep learning
- **Positional encoding in NeRF**
  - positional encoding fundamentally changes the function space that the MLP can represent
    - applying Fourier features changes the similarity measure between inputs
      - modifies the spectrum of the induced kernel
      - increases sensitivity to high-frequency variations
  - NeRF applies Fourier-based positional encoding to input coordinates
    - maps low-dimensional inputs into a high-dimensional feature space

InstantNGP

- accelerate neural field training and rendering (NeRF-style)
- represent coordinates with a multi-resolution hash grid instead of high-dim sinusoidal encoding

    - small MLP predicts density/color from hashed features
- hash grid gives strong spatial features early → optimization becomes easier
    - multi-resolution captures both coarse structure and fine details
- limitation
    - hash collisions can introduce artifacts
    - still requires ray marching for rendering

Gaussian Splatting

- goal: fast, high-quality novel view synthesis without heavy ray marching
- scene representation
    - represent a scene as a set of 3D Gaussians
    - each Gaussian has
        - position, covariance (shape), opacity, color (often with SH coefficients)
- rendering
    - differentiable splatting onto the image plane
    - alpha compositing in depth order
- why good
    - training is efficient and stable
    - rendering is real-time friendly (compared to NeRF)
- limitations
    - needs good initialization (from SfM/colmap-style points)
    - large scenes can require many Gaussians (memory)