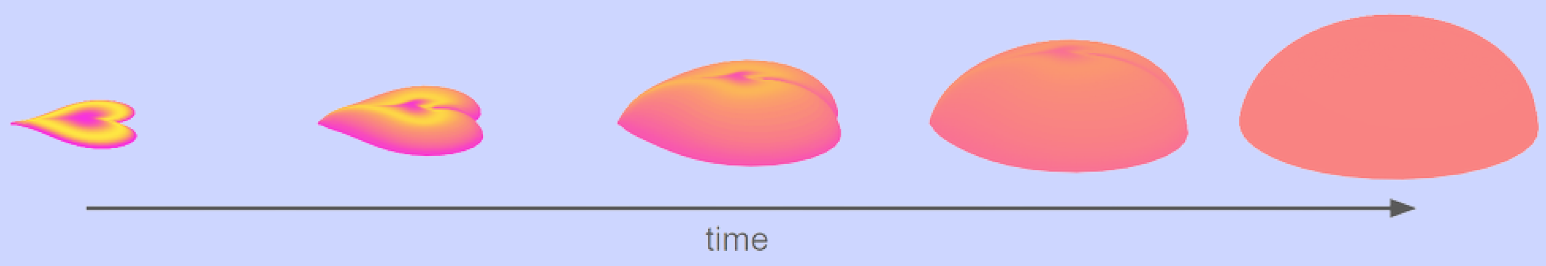
Investigating Poisson Flow Generative Models

UNIVERSITY OF OSLO

Background & Theory

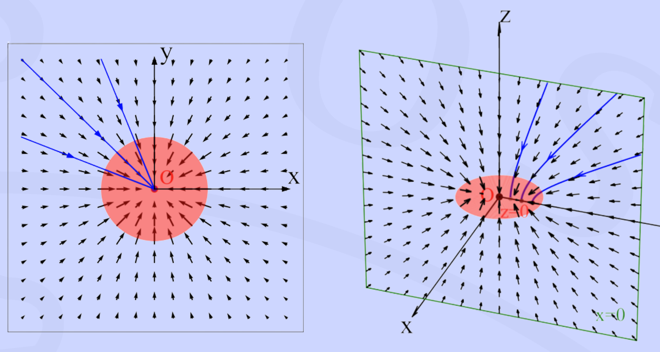


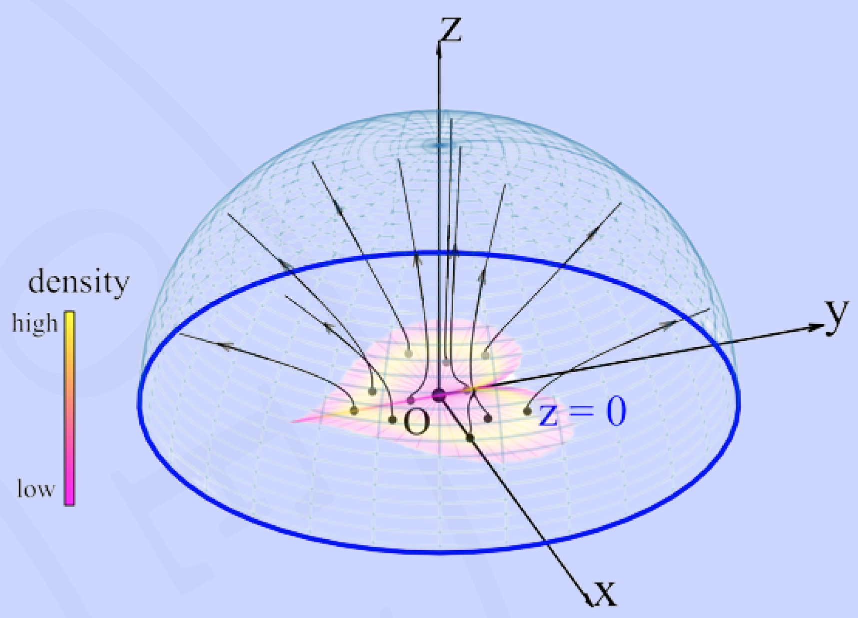
Overall Goal
Generate entirely new data (e.g. images)

Goal of PFGM
Generate faster and better than state of the art architectures (like diffusion models)

Difference from Conventional Diffusion

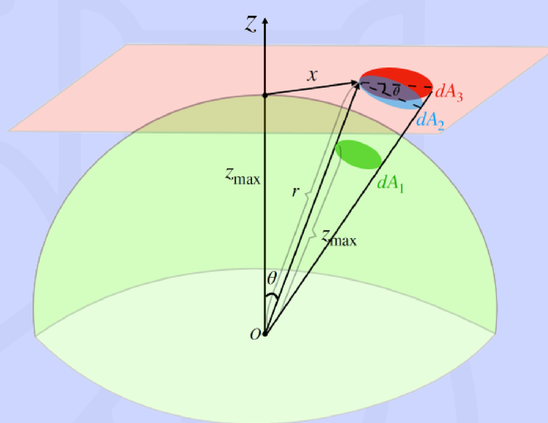
- Poissons equation replaces the Wiener process
- Uniform prior along the pixel values instead of Gaussian
- The negative Poisson field points to origin, resulting in mode collapse
  - Resolve by adding an intermittent channel





Hemispherical Uniform Prior

- Radial-directional decomposition of prior:
  - Projected isotropic Gaussian for bottom angle
  - Radius from origin to the plane to determine the remaining angle

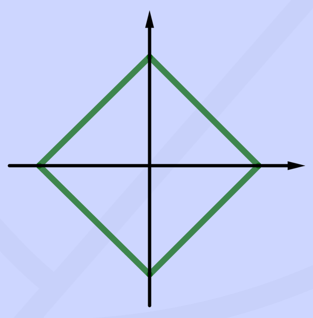


Next: Sparse Poisson Flow

Induce the Poisson flow to generate sparse discrete cosine transform (DCT) representations of the images

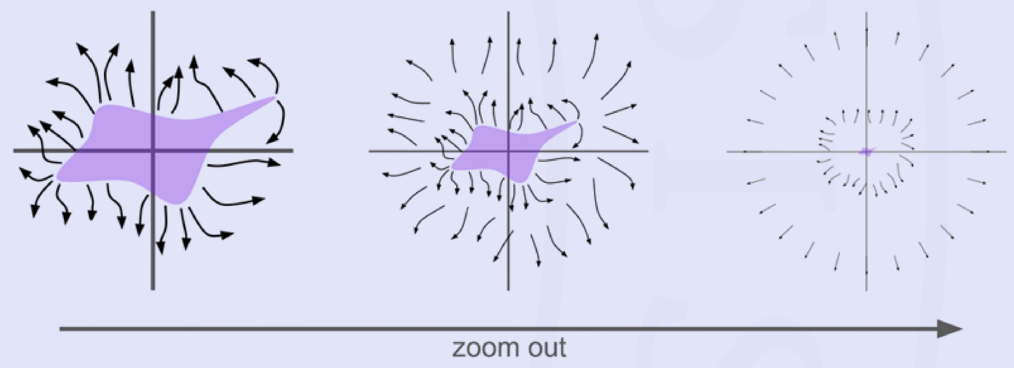
Altering the Hemispherical Uniform Prior:

- The angular component is replaced with an L-1 uniform sphere projected unto the L-2 sphere
  - In theory, this should induce sparsity analogously to lasso
- The radial component is kept identical
- Train the model on a sparsity matrix extracted from a DCT transformation of the images and then transform it back



We hypothesize this prior update to induce sparse learning, improving performance

Inspired by Physics



- Electric field (aka Poisson field) generated by distribution of charged particles results in uniform angular distribution at far enough distance
- Laws of physics provide invertible mapping between uniform distribution and data distribution
- Learn empirical Poisson field generated by training data distribution
- Train U-Net (DDPM++ backbone) to estimate empirical field based on a data point.
- Sample uniformly from high-dimensional hemisphere and run dynamics in reverse time to generate novel images from data distribution

Feedback on their Paper

- We gave the authors feedback on a typo confusing a random variable with its pdf. They responded that they will update the paper with a correction
- They are wrapping TensorFlow Tensors, inside PyTorch tensors

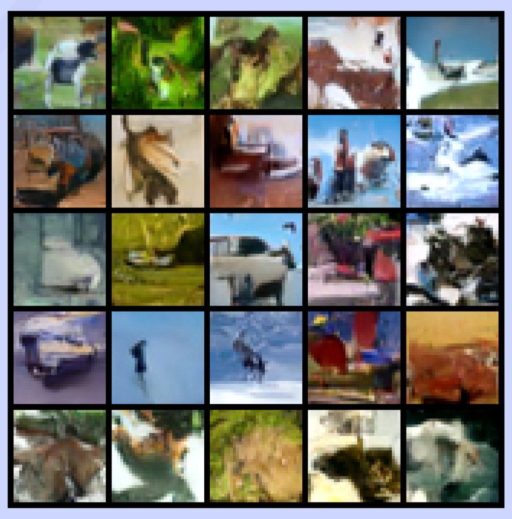
Feedback on Reproducibility

- We would love to see a conda environment, or a docker image
- The actual sampling procedure is explained differently than implemented

Provisional Results

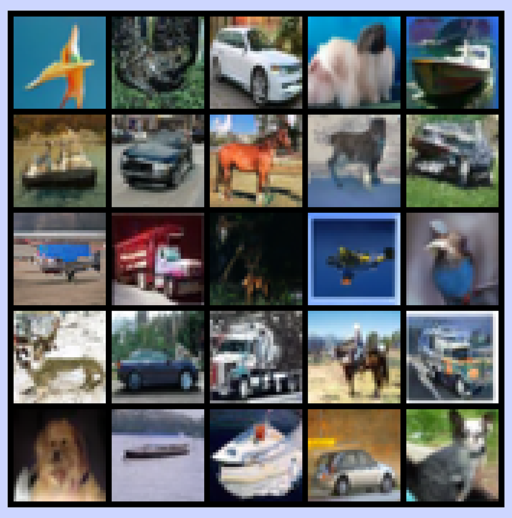
Image Generation (right)

- Sample from N^d hemisphere
- Use reverse dynamics (U-Net)



Interpolation (right bottom)

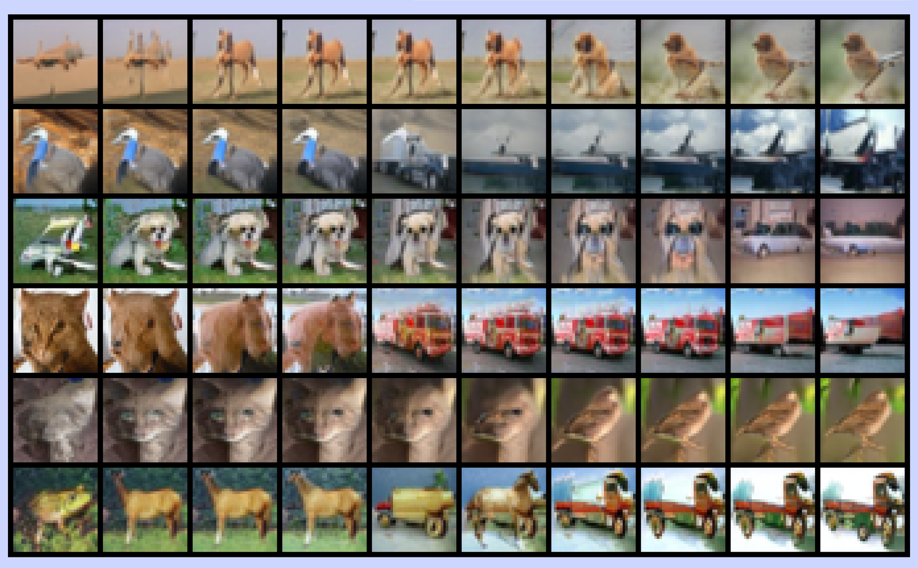
- Sample two points on hemisphere and interpolate
- Generate images from samples as before



Evaluation (below)

- Frechet Inception Distance (FID)
- Inception score
- More informative (vs loss)

Name	L_inf	L_2	L2-Paper
FID ↓	32.9	32.1	2.48
Inception ↑	9.03	8.9	9.65
Steps	130 000	130 000	1 300 000
Learning Rate	2e-4	2e-4	2e-4



Based on Xu, Yilun, et al. "Poisson flow generative models." Advances in Neural Information Processing Systems 35 (2022): 16782-16795.
Images from Paper and from: https://www.assemblyai.com/blog/an-introduction-to-poisson-flow-generative-models/