Principled Interpolation in Normalizing Flows

Samuel G. Fadel*   Sebastian Mair*   Ricardo da S. Torres   Ulf Brefeld

* Equal contribution

Leuphana University of Lüneburg, Germany

University of Campinas, Brazil

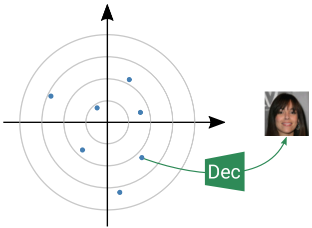
Norwegian University of Science and Technology, Norway

LEUPHANA UNIVERSITÄT LÜNEBURG   UNICAMP   NTNU

(Deep) Generative Models

Standard setting for VAEs, GANs, and Normalizing Flows:

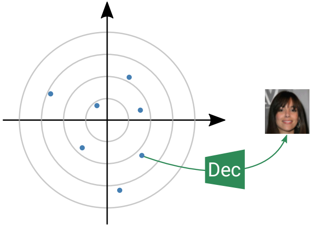
1. Assume a prior distribution over *latent* variables
2. **Decoder** generates data from **samples** of this distribution



(Deep) Generative Models

Standard setting for VAEs, GANs, and Normalizing Flows:

1. Assume a prior distribution over *latent* variables
2. **Decoder** generates data from **samples** of this distribution



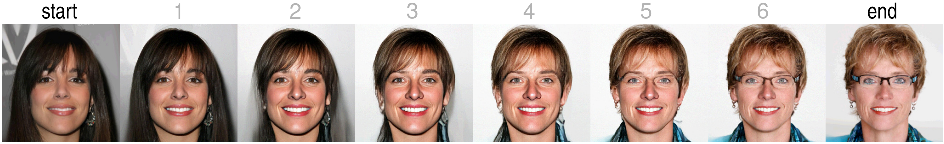
Generative models are often evaluated with **interpolations**

Interpolation in a Deep Generative Model

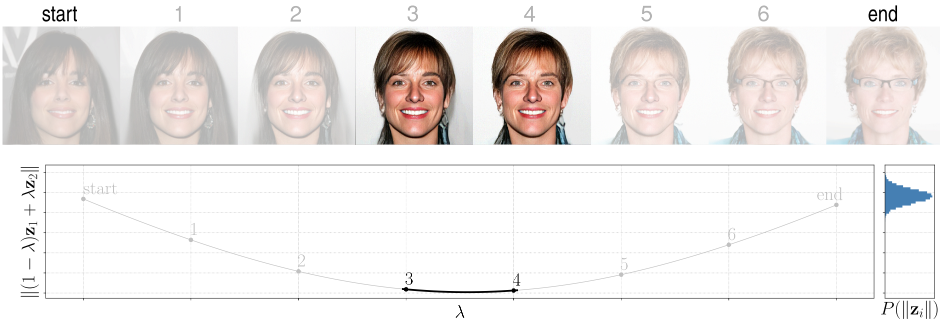
start

end

Interpolation in a Deep Generative Model

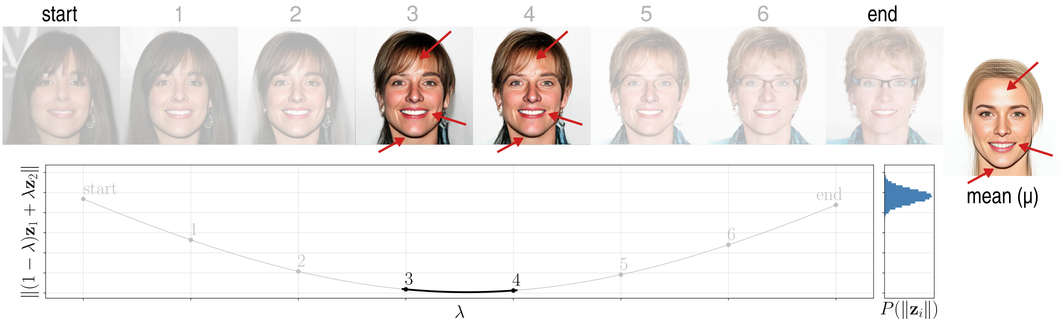


Interpolation in a Deep Generative Model

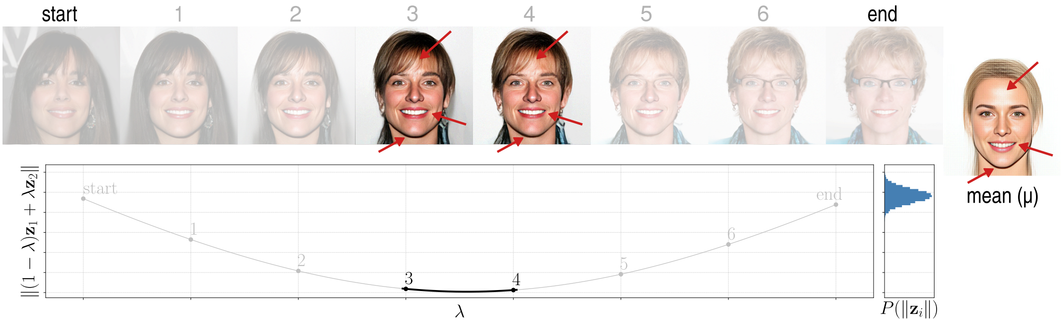


Norms should follow a χ-**distribution** due to a Gaussian prior

Interpolation in a Deep Generative Model



Interpolation in a Deep Generative Model



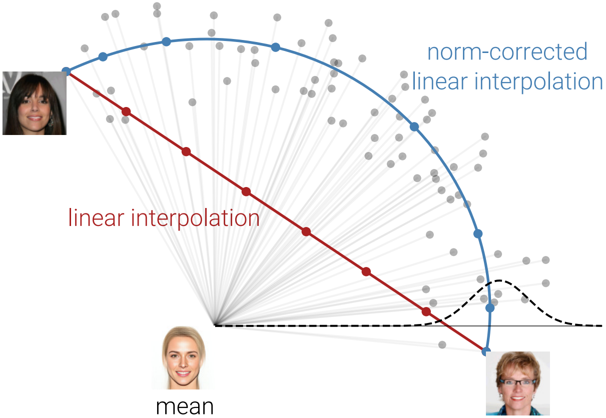
What if norms were corrected?

Interpolation in a Deep Generative Model

Interpolation in a Deep Generative Model

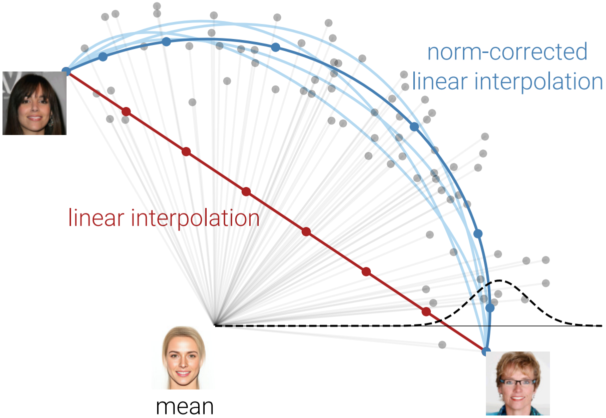
Norm correction **biases** interpolants towards start/end

Alternative Interpolations



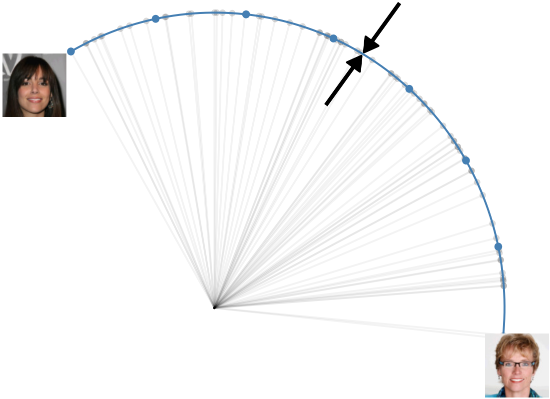
Alternative Interpolations

Which interpolation to pick from?



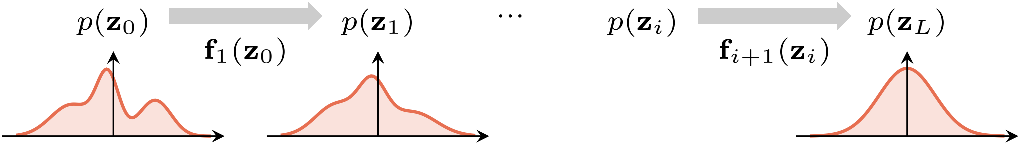
Alternative Interpolations

Make all norms **fixed**!



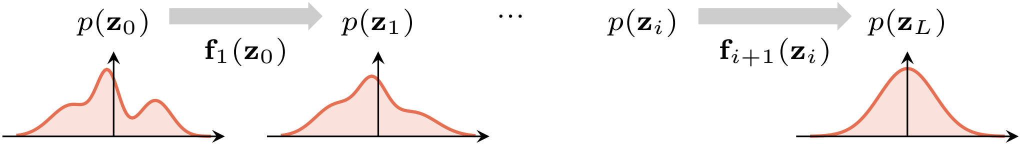
Normalizing Flows in a Nutshell

- By change of variable theorem: $p_x(\mathbf{x}) = p_z(f(\mathbf{x})) \cdot |\det J_f|$
- Chain of transformations $f = f_L \circ f_{L-1} \circ \cdots \circ f_1$



Normalizing Flows in a Nutshell

- By change of variable theorem: $p_x(\mathbf{x}) = p_z(f(\mathbf{x})) \cdot |\det J_f|$
- Chain of transformations $f = f_L \circ f_{L-1} \circ \cdots \circ f_1$



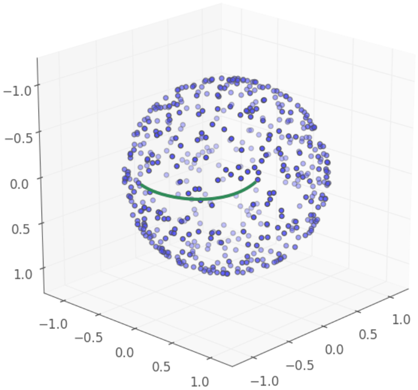
- Training parameters of f via maximum likelihood
- This defines a generative model:
  1. Sample $z \sim p_z(z)$ (usually a Gaussian)
  2. Obtain $x = f^{-1}(z)$ via inverse transformation

Proposed Approach

- Normalizing Flow to p-norm sphere
- Example: $p = 2$ yields a hypersphere
- **Prior distribution:** von Mises-Fisher
- **Interpolation:** spherical linear

$$\gamma(\lambda) = \frac{\sin((1-\lambda)\omega)}{\sin(\omega)}\mathbf{s}_a + \frac{\sin(\lambda\omega)}{\sin(\omega)}\mathbf{s}_b,$$

where ω is the angle between $\mathbf{s}_a$ and $\mathbf{s}_b$

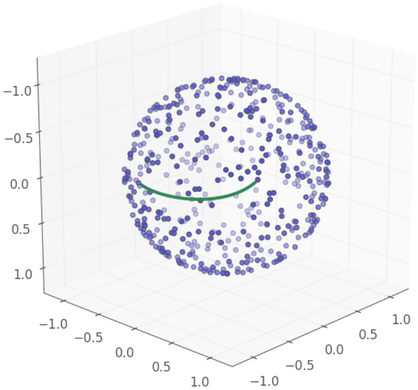


Proposed Approach

- Normalizing Flow to p-norm sphere
- Example: $p = 2$ yields a hypersphere
- **Prior distribution:** von Mises-Fisher
- **Interpolation:** spherical linear

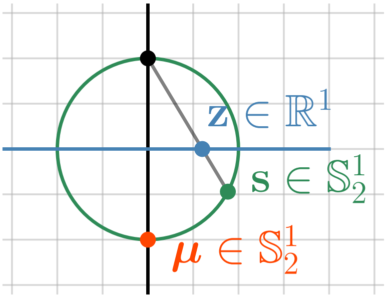
$$\gamma(\lambda) = \frac{\sin((1-\lambda)\omega)}{\sin(\omega)}\mathbf{s}_a + \frac{\sin(\lambda\omega)}{\sin(\omega)}\mathbf{s}_b,$$

where ω is the angle between $\mathbf{s}_a$ and $\mathbf{s}_b$



How to move from $\mathbb{R}^d$ to a hypersphere?

Proposed Approach

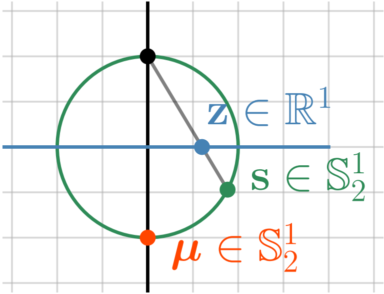


- Let $\mathbf{z}$ be the representation after f_L
- We use a stereographic projection

$$\psi(\mathbf{z}) = \mathbf{s} = \begin{bmatrix} \mathbf{z}\rho_{\mathbf{z}} \\ 1 - \rho_{\mathbf{z}} \end{bmatrix}, \text{ with } \rho_{\mathbf{z}} = \frac{2}{1 + \|\mathbf{z}\|^2}$$

to move $\mathbf{z}$ to the **hypersphere**

Proposed Approach



- Let $\mathbf{z}$ be the representation after f_L
- We use a stereographic projection

$$\psi(\mathbf{z}) = \mathbf{s} = \begin{bmatrix} \mathbf{z}\rho_{\mathbf{z}} \\ 1 - \rho_{\mathbf{z}} \end{bmatrix}, \text{ with } \rho_{\mathbf{z}} = \frac{2}{1 + \|\mathbf{z}\|^2}$$

to move $\mathbf{z}$ to the **hypersphere**

- ! bijective almost everywhere
- ✓ efficient Jacobian
- ✓ no additional parameters

Quantitative Results on Data Generation

	Prior	Test BPD	Sample FID
Fashion MNIST	Gaussian	3.24 ± 0.04	66.64 ± 1.29
	vMF $\kappa = 1d$	**3.16** ± 0.03	**60.45** ± 3.34
	vMF $\kappa = 1.5d$	3.30 ± 0.07	61.89 ± 1.29
	vMF $\kappa = 2d$	3.22 ± 0.06	60.60 ± 3.47
CIFAR10	Gaussian	3.52 ± 0.01	71.34 ± 0.45
	vMF $\kappa = 1d$	3.43 ± 0.00	71.07 ± 0.78
	vMF $\kappa = 1.5d$	**3.42** ± 0.00	**70.58** ± 0.40
	vMF $\kappa = 2d$	**3.42** ± 0.01	71.00 ± 0.28

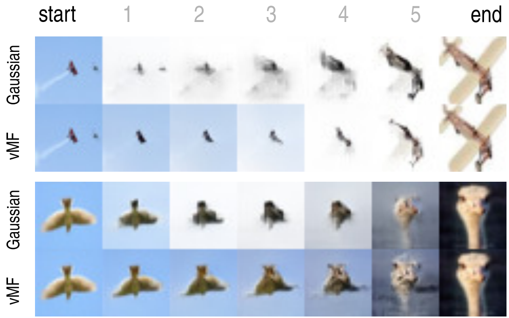
- Our approach has **competitive** generative performance

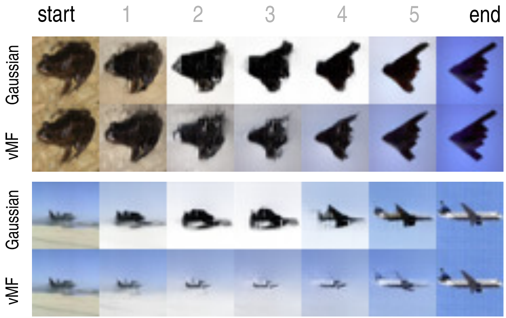
Quantitative Results on Interpolation

	Prior	Interpolation	BPD	FID
Fashion MNIST	Gaussian	linear	2.84 ± 0.10	13.06 ± 0.62
	Gaussian	norm-corr. linear	2.93 ± 0.03	**7.80** ± 0.13
	vMF $\kappa = 1d$	spherical	**2.66** ± 0.03	**12.16** ± 0.13
	vMF $\kappa = 1.5d$	spherical	2.84 ± 0.07	12.19 ± 1.07
	vMF $\kappa = 2d$	spherical	2.70 ± 0.05	15.11 ± 0.85
CIFAR10	Gaussian	linear	2.64 ± 0.06	58.63 ± 1.26
	Gaussian	norm-corr. linear	3.32 ± 0.01	**14.29** ± 0.16
	vMF $\kappa = 1d$	spherical	2.78 ± 0.05	**51.08** ± 0.37
	vMF $\kappa = 1.5d$	spherical	2.66 ± 0.05	55.23 ± 5.14
	vMF $\kappa = 2d$	spherical	**2.58** ± 0.08	52.65 ± 3.34

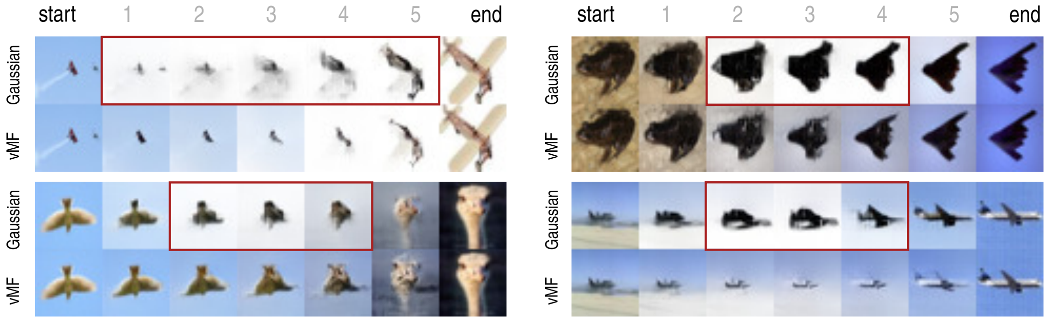
- Gaussian with linear interpolation is consistently worst
- Norm-corrected interpolation is **biased** towards real data
- Proposed approach yields **better** results for all metrics

Qualitative Results on Interpolation





Qualitative Results on Interpolation



- Gaussian **loses background** due to the mean image
- Proposed approach yields **smoother** interpolations

Summary

- ✗ Gaussian has issues with linear interpolations and yields biased interpolations
- ! We suggest intuitive, unique paths on hyperspheres
- ✓ Approach maintains generative performance with better interpolations

Check out our paper: