Invertible Monotone Operators for Normalizing Flows

NeurIPS 2022

Byeongkeun Ahn, Chiyoon Kim, Youngjoon Hong, Hyunwoo J. Kim

MLV Machine Learning & Vision Lab

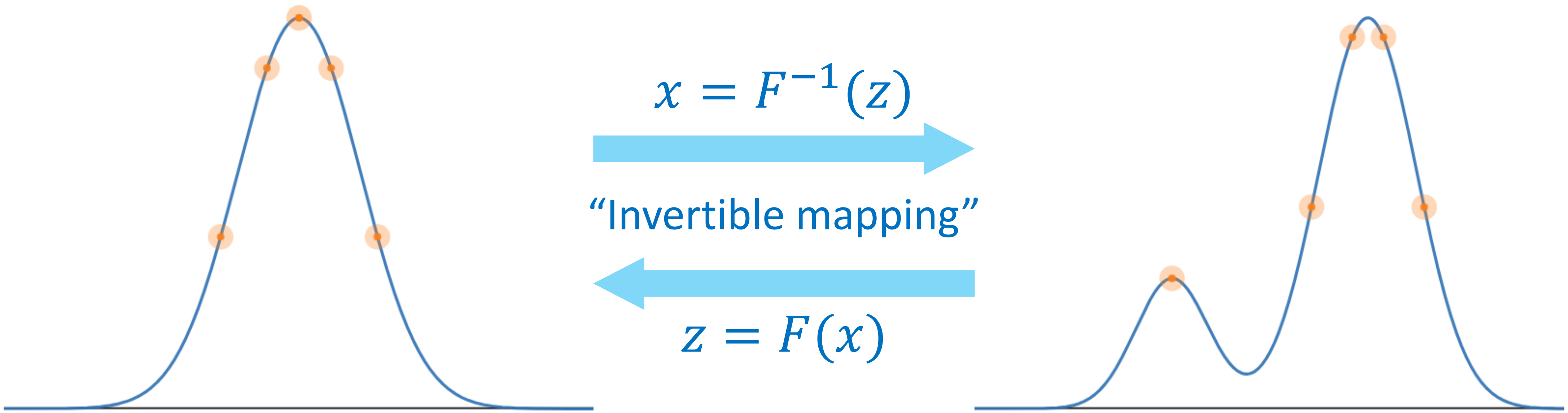
KOREA UNIVERSITY

SUNGKYUNKWAN UNIVERSITY (SKKU)

Normalizing Flows (NFs)

Base distribution

Target distribution



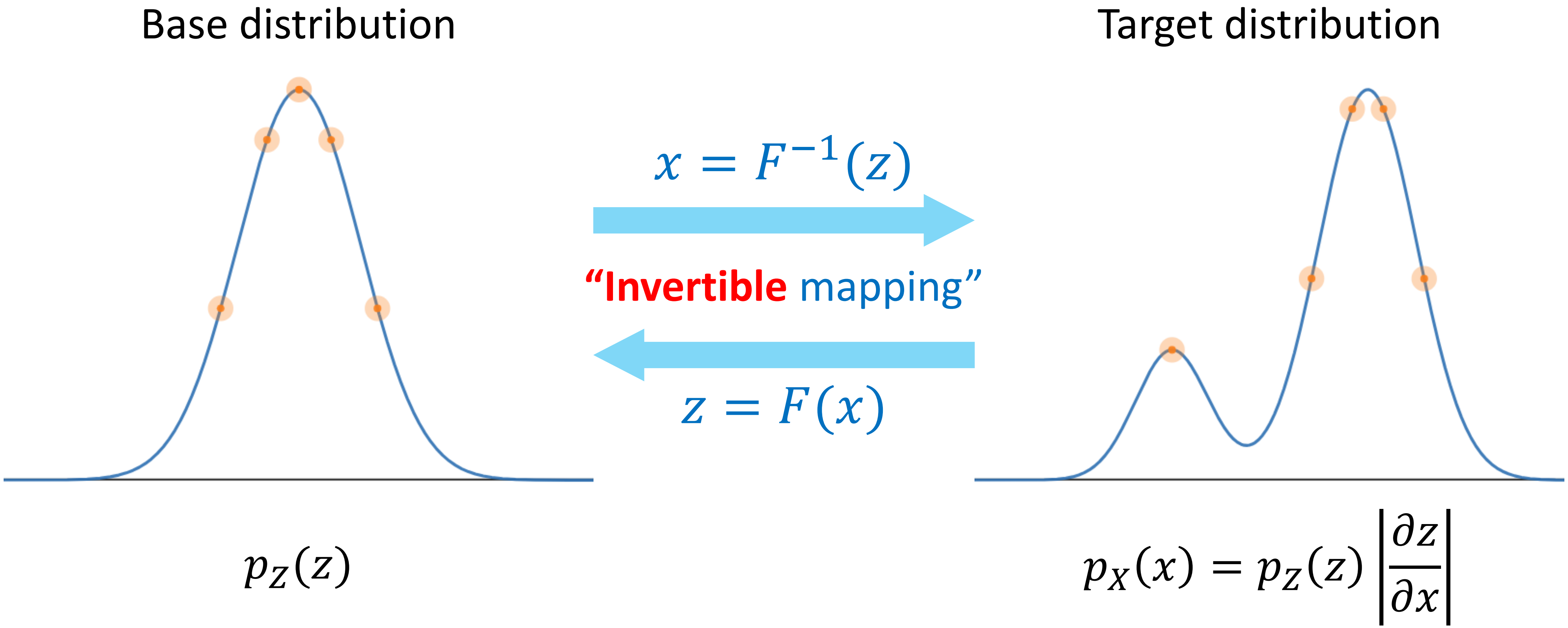
$p_Z(z)$

$p_X(x) = p_Z(z) \left| \frac{\partial z}{\partial x} \right|$

$$\log p_X(x) = \log p_Z(z) + \log \det J_F$$

* $J_F = \partial z / \partial x$ is the Jacobian of F at point x

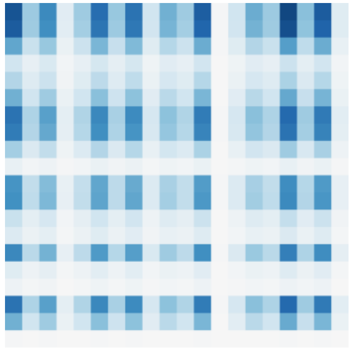
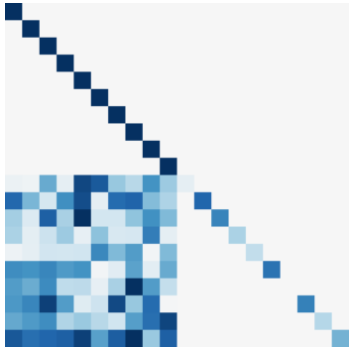
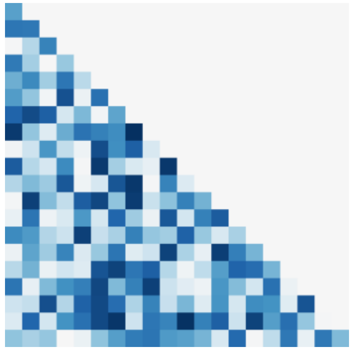
Normalizing Flows (NFs)



$$\log p_X(x) = \log p_Z(z) + \mathbf{\log \det} \boldsymbol{J_F}$$

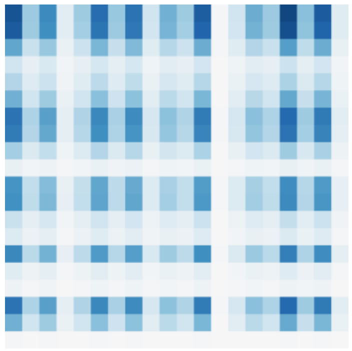
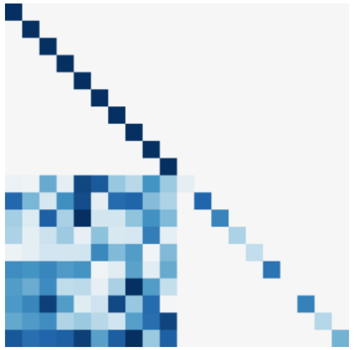
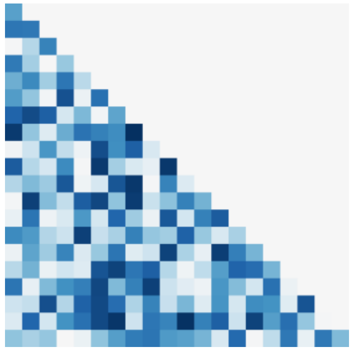
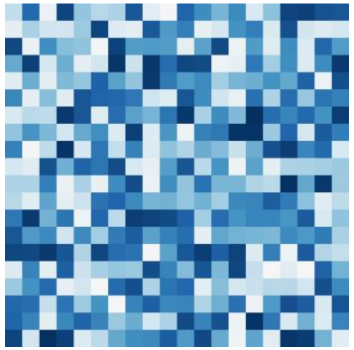
* $J_F = \partial z / \partial x$ is the Jacobian of F at point x

Limitation of Existing Architectures

	Low-rank factorizations	Coupling blocks	Autoregressive structures
Jacobian			
Structure	Constrained	Constrained	Constrained

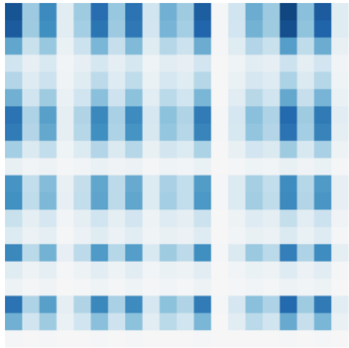
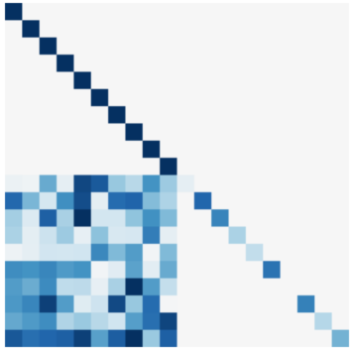
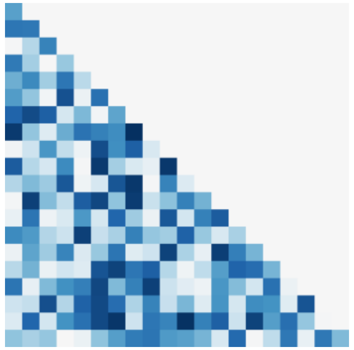
* Images were adopted from Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.

Limitation of Existing Architectures

	Low-rank factorizations	Coupling blocks	Autoregressive structures	ResNet-based
Jacobian				
Structure	Constrained	Constrained	Constrained	Not constrained

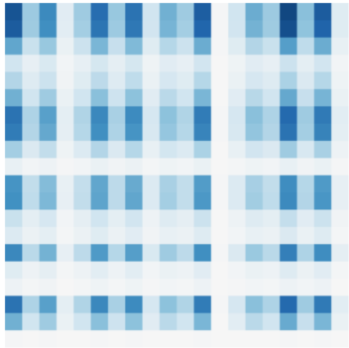
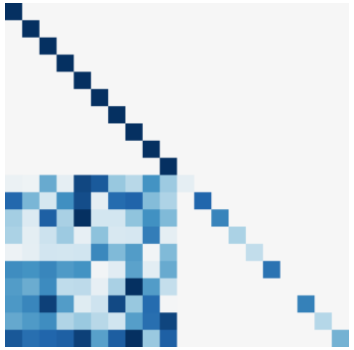
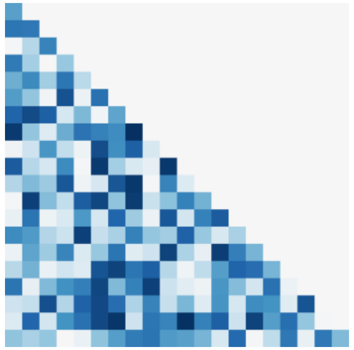
* Images were adopted from Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.

Limitation of Existing Architectures

	Low-rank factorizations	Coupling blocks	Autoregressive structures	ResNet-based
Jacobian				
Structure	Constrained	Constrained	Constrained	Not constrained
Lipschitz constant	Not constrained	Not constrained	Not constrained	Constrained

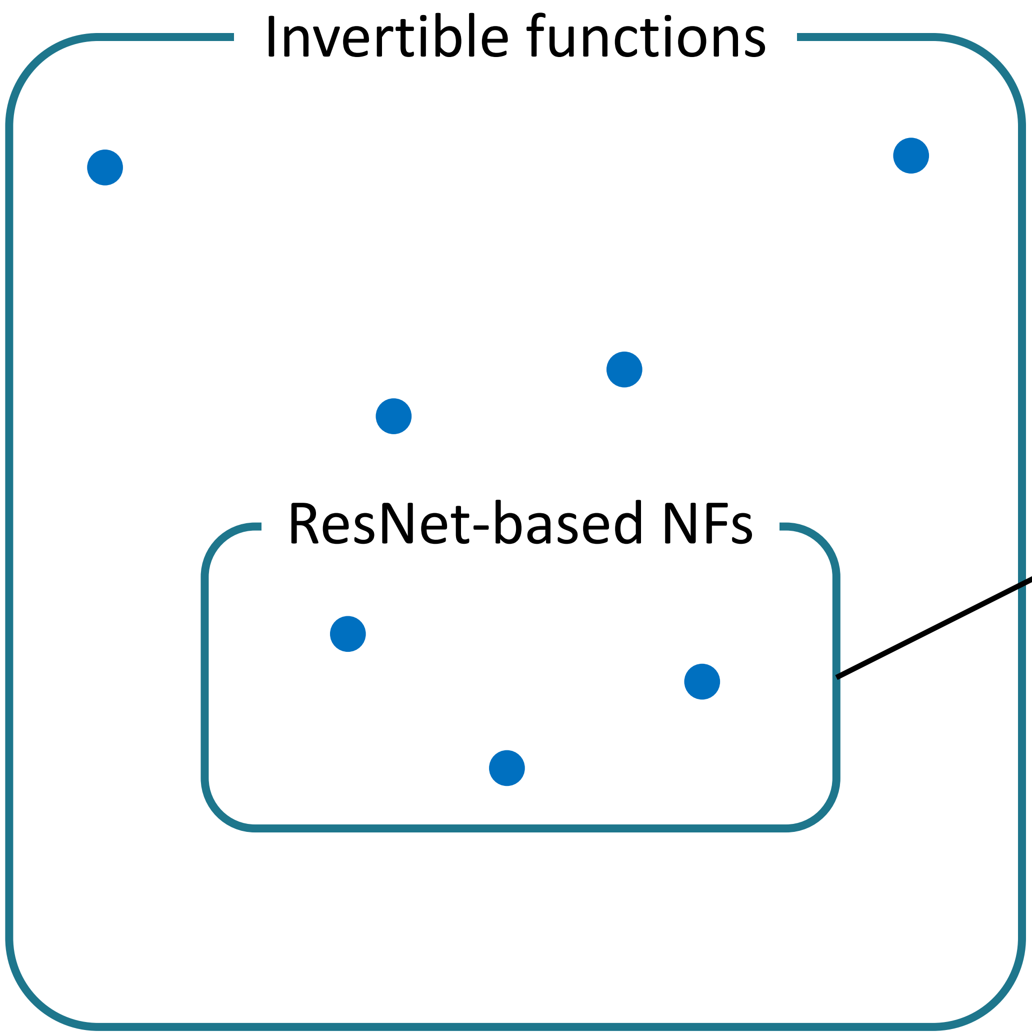
* Images were adopted from Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.

Limitation of Existing Architectures

	Low-rank factorizations	Coupling blocks	Autoregressive structures	ResNet-based	**Monotone Flows**
Jacobian					
Structure	Constrained	Constrained	Constrained	Not constrained	**Not constrained**
Lipschitz constant	Not constrained	Not constrained	Not constrained	Constrained	**Not constrained**

* Images were adopted from Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.

A Crucial Observation



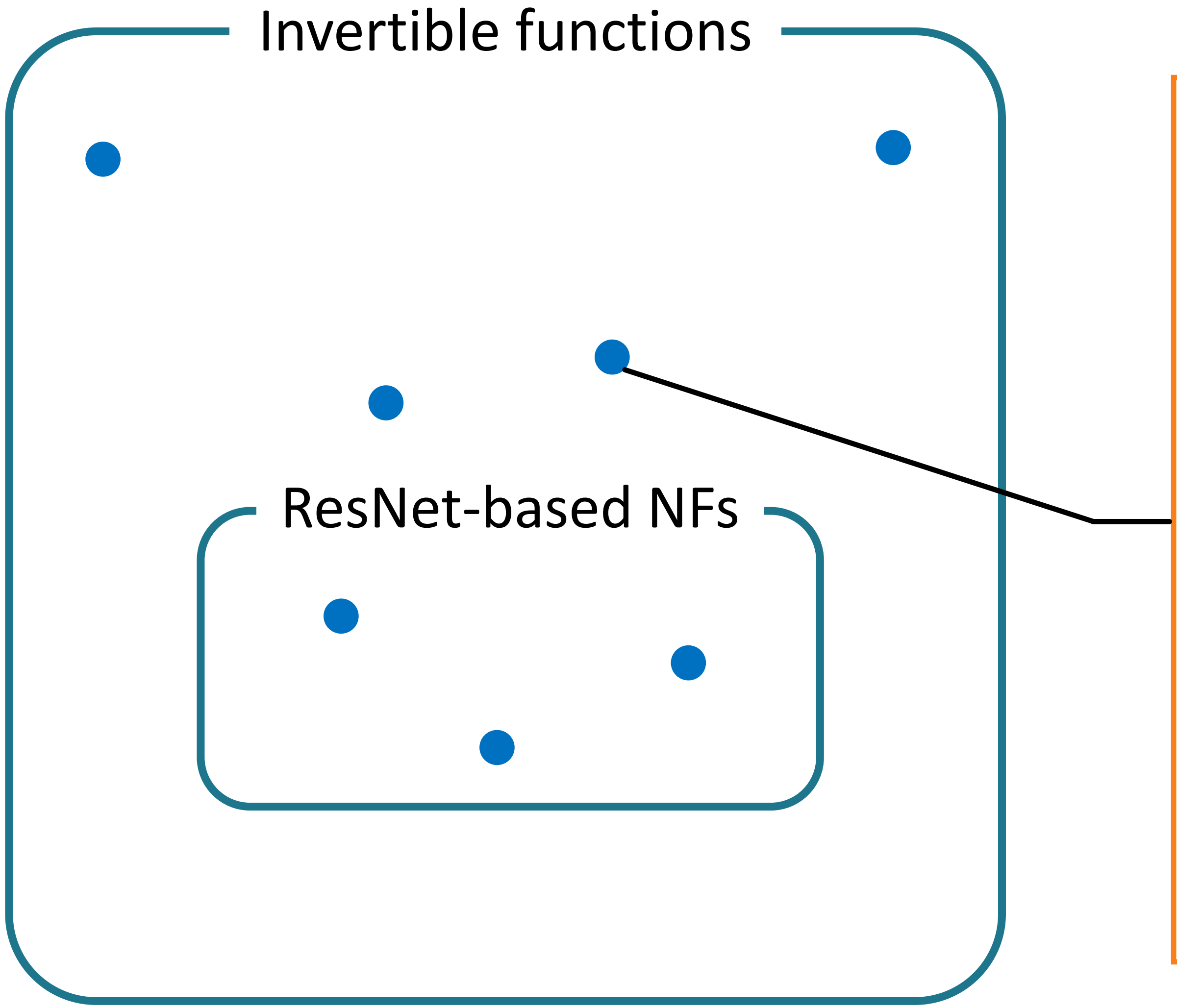
How do ResNet-based NFs work?

- $F(x) = x + G(x)$ with $\mathrm{Lip}(G) < 1$ (i.e., G is a contraction mapping)

$$y = F(x) \leftrightarrow x = y - G(x)$$

→ contraction mapping
→ unique fixed point

A Crucial Observation



A counterexample

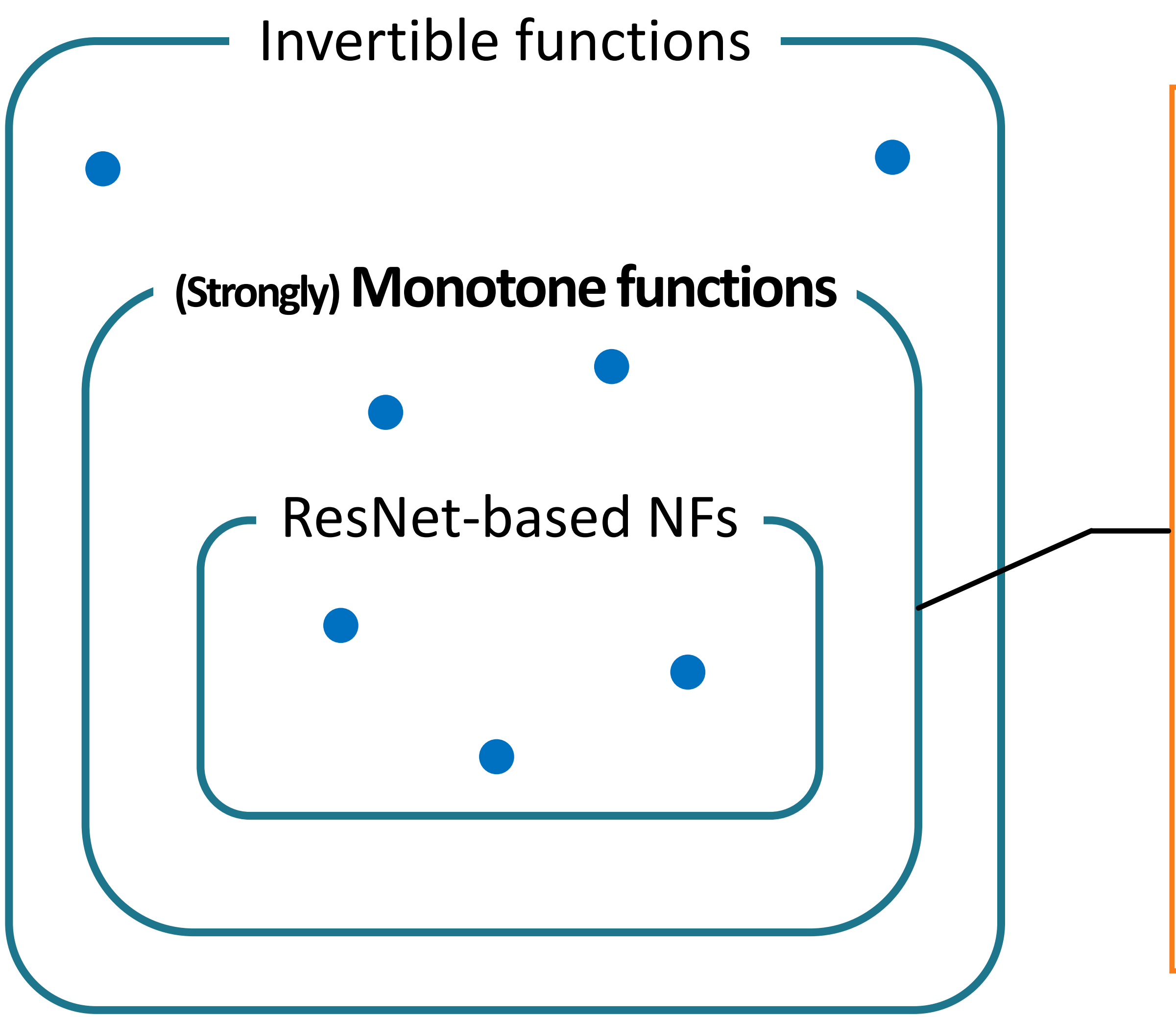
- $F(x) = x + G(x)$ with $G(x) = 5x$

  → G is expansive $(\mathrm{Lip}(G) = 5 > 1)$

  → But $F(x) = 6x$ is invertible

  → G does NOT need to be a contraction to ensure F is invertible!

A Crucial Observation



ResNet-based NFs are strongly monotone functions

- $F(x) = x + G(x)$ with $\mathrm{Lip}(G) < 1$ (i.e., G is a contraction mapping)

$$\rightarrow \frac{\mathrm{d}F(x)}{\mathrm{d}x} \geq 1 - \mathrm{Lip}(G) > 0$$

: invertible

Monotone Functions in $\mathbb{R}^n$

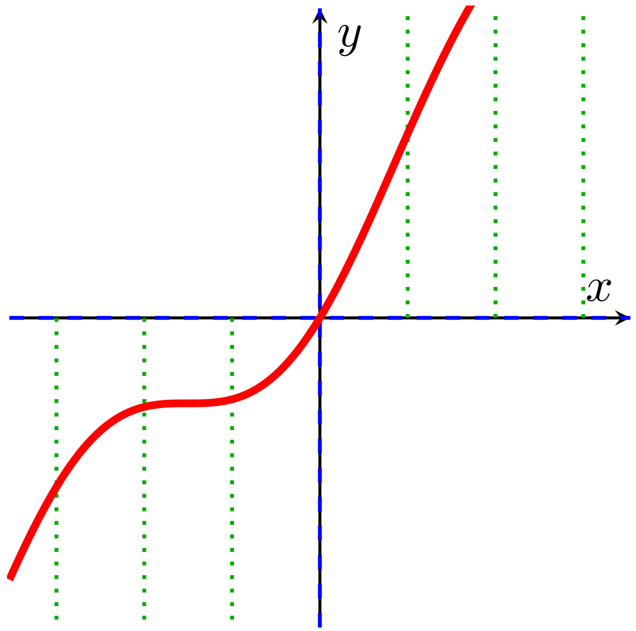
- In $\mathbb{R}$, a function $F\colon\ \mathbb{R} \to \mathbb{R}$ is monotone
  $\Leftrightarrow$ $x < y$ implies $F(x) \leq F(y)$
  $\Leftrightarrow$ $\big(F(x) - F(y)\big)(x - y) \geq 0$ for all $x, y \in \mathbb{R}$

- In $\mathbb{R}^n$, a function $F\colon\ \mathbb{R}^n \to \mathbb{R}^n$ is monotone
  $\Leftrightarrow$ $\langle F(x) - F(y),\ x - y\rangle \geq 0$ for all $x, y \in \mathbb{R}^n$

* $\langle \cdot, \cdot \rangle$ denotes a dot product

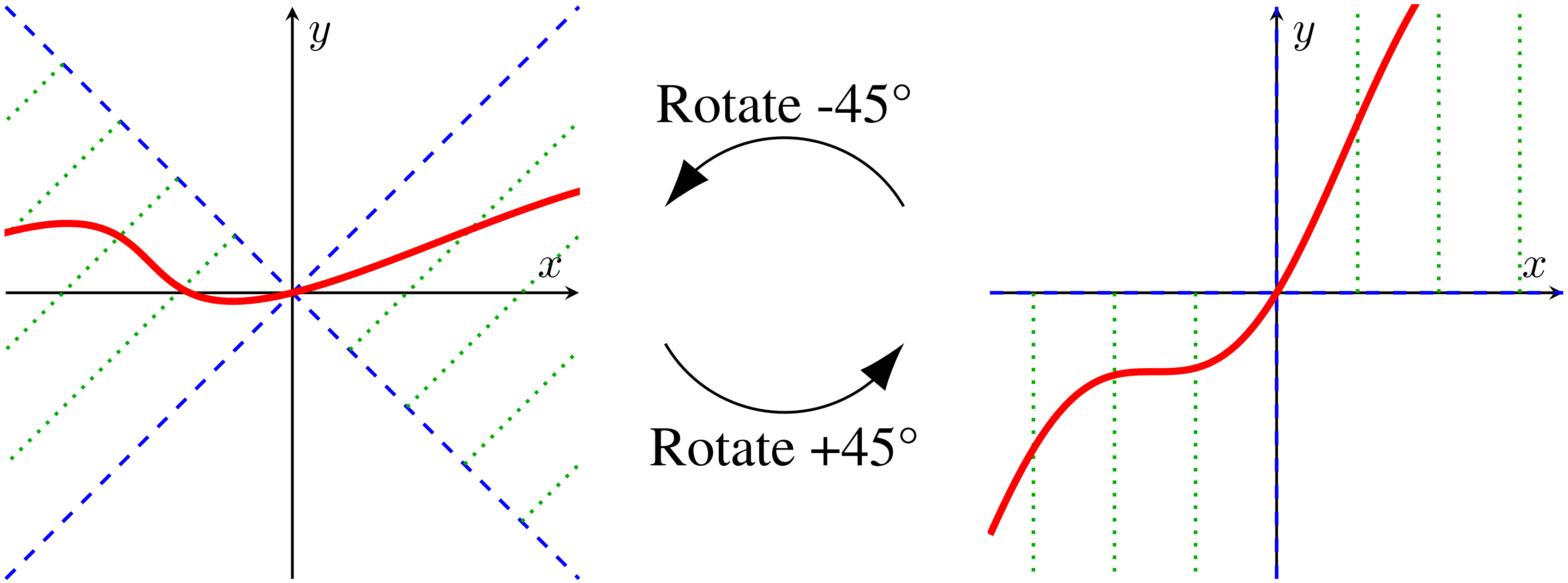
The Geometric Construction

Monotone operators

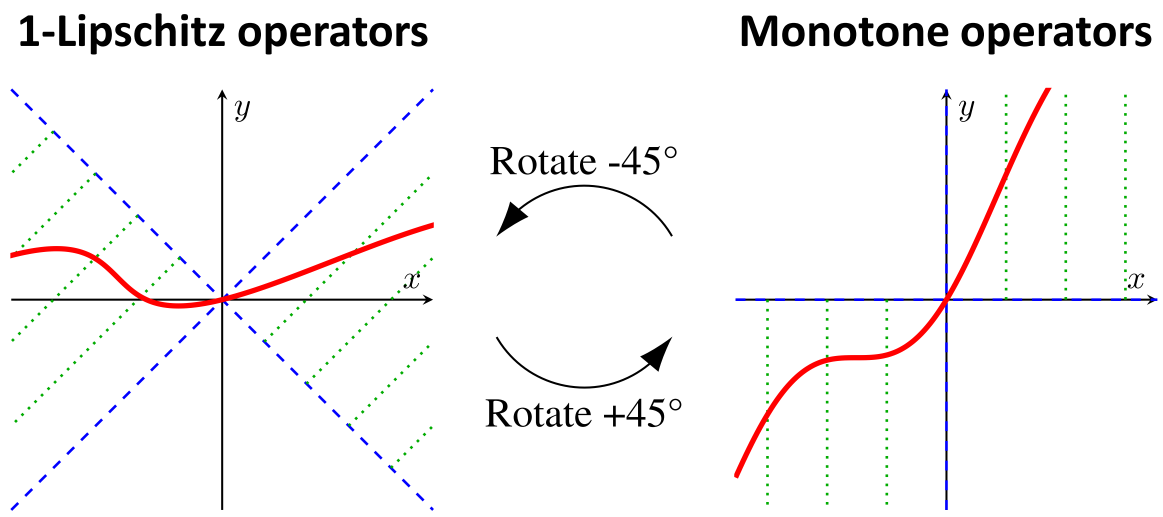


The Geometric Construction

1-Lipschitz operators **Monotone operators**



The Geometric Construction



- For an L-Lipschitz function G ($L < 1$), the monotone formulation is defined as

$$F(x) = \left(\frac{\mathrm{Id} + G}{2}\right)^{-1}(x) - x$$

- The inverse resembles very much the forward computation:

$$F^{-1}(y) = \left(\frac{\mathrm{Id} - G}{2}\right)^{-1}(y) - y$$

* F and the inverse F^{-1} are well-defined when G has a Lipschitz constant $L < 1$, because F and F^{-1} become strongly monotone.

Training Algorithm

- Training objective: maximum likelihood

- Backpropagation through the inverse of $(\mathrm{Id} + G)$: *

$$w = (\mathrm{Id} + G)^{-1}(u) \qquad \frac{\partial \ell}{\partial u} = \frac{\partial \ell}{\partial w}(I + J_G)^{-1}, \quad \frac{\partial \ell}{\partial \theta} = \left(\frac{\partial \ell}{\partial w}(I + J_G)^{-1}\right)\frac{\partial G}{\partial \theta}$$

- Log-determinant computation: **

$$\log\det J_F = \mathrm{tr}[\log(I - J_G) - \log(I + J_G)] = \mathbb{E}_{n\sim p_N(n), v\sim\mathcal{N}(0,I)}\left[\sum_{k=1}^{n} \frac{(-1) - (-1)^{k+1}}{k} \frac{v^T J_G^k v}{P(N \geq k)}\right]$$

where J_G is evaluated at $w = \left(\frac{\mathrm{Id}+G}{2}\right)^{-1}(x)$.

* Adapted from Lu et al., Implicit Normalizing Flows, ICLR 2021.

** Adapted from Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.

Concatenated Pila

$$\mathrm{Pila}(x) = \begin{cases} x & \text{if } x \geq 0, \\ \left(\frac{k^2}{2}x^3 - kx^2 + x\right)e^{kx} & \text{if } x < 0. \end{cases} \qquad k = 5$$

$$\mathrm{CPila}(x) = \alpha_1[\mathrm{Pila}(x - \alpha_2), \mathrm{Pila}(-x - \alpha_2)]^T \qquad \text{where } \alpha_1 = 1/1.06 \text{ and } \alpha_2 = 0.2.$$

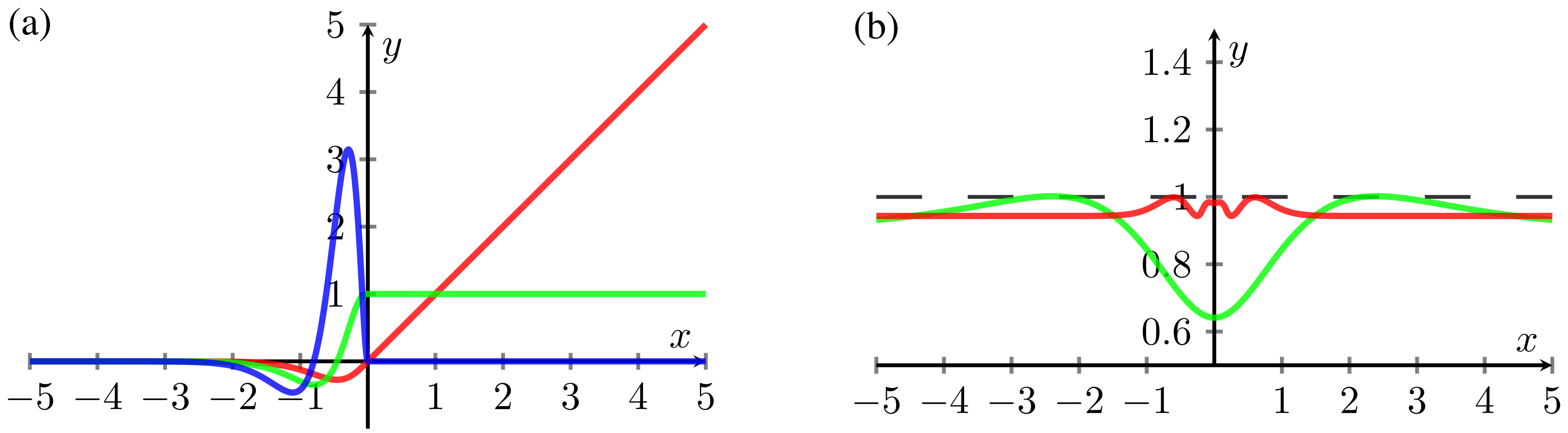


Figure 2: Graphical illustrations of Pila and CPila. (a) The graph of Pila (red) and its first (green) and second derivatives (blue) with $k = 5$. (b) The speed of the curve of CPila (red) with $k = 5$ and CLipSwish (green) with $\beta = 1$.

Theoretical Result

- Monotone Flows have a strictly better expressive power!

Definition 3. *For* $0 \leq L < 1$,

L-Lipschitz functions	$\mathcal{G}_L = \left\{G \in C^2(\mathbb{R}^n, \mathbb{R}^n) \middle| \mathrm{Lip}(G) = L\right\}$
Residual formulation*	$\mathcal{R}_L = \{\mathrm{Id} + G \mid G \in \mathcal{G}_L\}$
Inverse residual formulation**	$\mathcal{I}_L = \left\{(\mathrm{Id} + G)^{-1} \middle| G \in \mathcal{G}_L\right\}$
Monotone formulation	$\mathcal{M}_L = \left\{\left(\frac{\mathrm{Id}+G}{2}\right)^{-1} - \mathrm{Id} \middle| G \in \mathcal{G}_L\right\}$

* Behrmann et al., Invertible Residual Networks, ICML 2019.
Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.
** Lu et al., Implicit Normalizing Flows, ICLR 2021. Note that in their work one implicit block is defined as the composition of one residual formulation and one inverse residual formulation; we analyze a smaller unit without loss of generality.

Theoretical Result

- Monotone Flows have a strictly better expressive power!

Definition 3. *For* $0 \leq L < 1$,

L-Lipschitz functions	$\mathcal{G}_L = \left\{G \in C^2(\mathbb{R}^n, \mathbb{R}^n) \mid \mathrm{Lip}(G) = L\right\}$
Residual formulation*	$\mathcal{R}_L = \{\mathrm{Id} + G \mid G \in \mathcal{G}_L\}$
Inverse residual formulation**	$\mathcal{I}_L = \left\{(\mathrm{Id} + G)^{-1} \mid G \in \mathcal{G}_L\right\}$
Monotone formulation	$\mathcal{M}_L = \left\{\left(\frac{\mathrm{Id}+G}{2}\right)^{-1} - \mathrm{Id} \mid G \in \mathcal{G}_L\right\}$

Theorem 4. *For* $0 \leq L < 1$,

(i) $\mathcal{I}_L = \frac{1}{1-L^2}\mathcal{R}_L$, *(ii)* $\mathcal{M}_L = \frac{1+L^2}{1-L^2}\mathcal{R}_{\frac{2L}{1+L^2}}$, *(iii)* $\mathcal{R}_L \subsetneq \mathcal{M}_L$, *(iv)* $\mathcal{I}_L \subsetneq \mathcal{M}_L$.

* Behrmann et al., Invertible Residual Networks, ICML 2019.
Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.
** Lu et al., Implicit Normalizing Flows, ICLR 2021. Note that in their work one implicit block is defined as the composition of one residual formulation and one inverse residual formulation; we analyze a smaller unit without loss of generality.

Theoretical Result

- Monotone Flows have a strictly better expressive power!

Definition 3. *For* $0 \leq L < 1$,

L-Lipschitz functions	$\mathcal{G}_L = \left\{G \in C^2(\mathbb{R}^n, \mathbb{R}^n) \mid \mathrm{Lip}(G) = L\right\}$
Residual formulation*	$\mathcal{R}_L = \{\mathrm{Id} + G \mid G \in \mathcal{G}_L\}$
Inverse residual formulation**	$\mathcal{I}_L = \left\{(\mathrm{Id} + G)^{-1} \mid G \in \mathcal{G}_L\right\}$
Monotone formulation	$\mathcal{M}_L = \left\{\left(\frac{\mathrm{Id}+G}{2}\right)^{-1} - \mathrm{Id} \,\middle|\, G \in \mathcal{G}_L\right\}$

Theorem 4. *For* $0 \leq L < 1$,

(i) $\mathcal{I}_L = \frac{1}{1-L^2}\mathcal{R}_L$, *(ii)* $\mathcal{M}_L = \frac{1+L^2}{1-L^2}\mathcal{R}_{\frac{2L}{1+L^2}}$, *(iii)* $\mathcal{R}_L \subsetneq \mathcal{M}_L$, *(iv)* $\mathcal{I}_L \subsetneq \mathcal{M}_L$.

* Behrmann et al., Invertible Residual Networks, ICML 2019.
Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.
** Lu et al., Implicit Normalizing Flows, ICLR 2021. Note that in their work one implicit block is defined as the composition of one residual formulation and one inverse residual formulation; we analyze a smaller unit without loss of generality.

Theoretical Result

- Monotone Flows have a strictly better expressive power!

Definition 3. *For* $0 \leq L < 1$,

L-Lipschitz functions	$\mathcal{G}_L = \left\{G \in C^2(\mathbb{R}^n, \mathbb{R}^n) \mid \mathrm{Lip}(G) = L\right\}$
Residual formulation*	$\mathcal{R}_L = \{\mathrm{Id} + G \mid G \in \mathcal{G}_L\}$
Inverse residual formulation**	$\mathcal{I}_L = \left\{(\mathrm{Id} + G)^{-1} \mid G \in \mathcal{G}_L\right\}$
Monotone formulation	$\mathcal{M}_L = \left\{\left(\frac{\mathrm{Id}+G}{2}\right)^{-1} - \mathrm{Id} \,\middle|\, G \in \mathcal{G}_L\right\}$

Theorem 4. *For* $0 \leq L < 1$,

(i) $\mathcal{I}_L = \frac{1}{1-L^2}\mathcal{R}_L$, *(ii)* $\mathcal{M}_L = \frac{1+L^2}{1-L^2}\mathcal{R}_{\frac{2L}{1+L^2}}$, *(iii)* $\mathcal{R}_L \subsetneq \mathcal{M}_L$, *(iv)* $\mathcal{I}_L \subsetneq \mathcal{M}_L$.

* Behrmann et al., Invertible Residual Networks, ICML 2019.
Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.
** Lu et al., Implicit Normalizing Flows, ICLR 2021. Note that in their work one implicit block is defined as the composition of one residual formulation and one inverse residual formulation; we analyze a smaller unit without loss of generality.

Theoretical Result

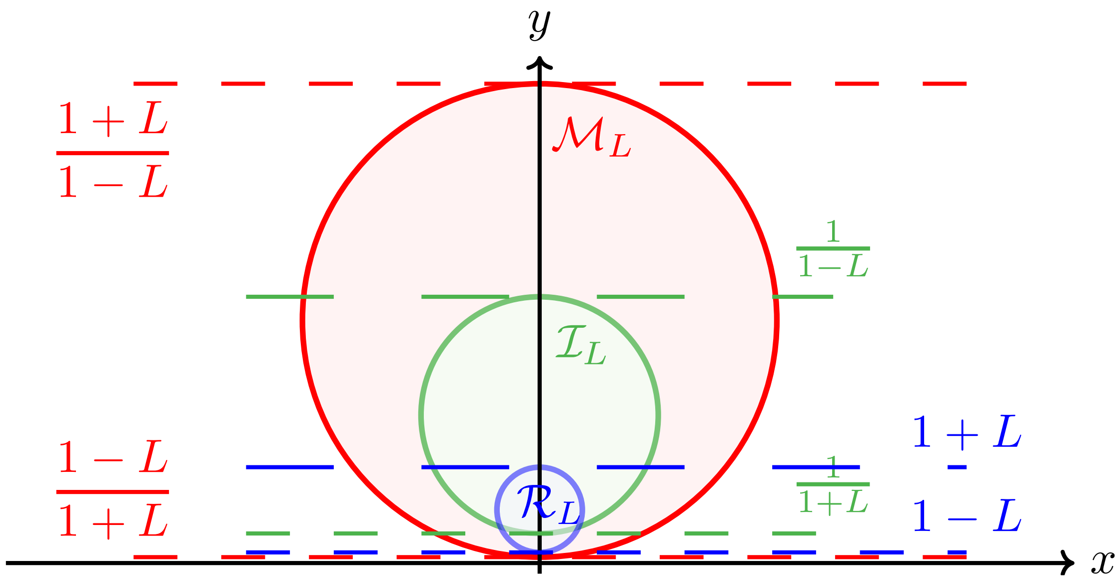
- Monotone Flows have a strictly better expressive power!

Definition 3. *For* $0 \leq L < 1$,

L-Lipschitz functions	$\mathcal{G}_L = \{G \in C^2(\mathbb{R}^n, \mathbb{R}^n) \mid \mathrm{Lip}(G) = L\}$
Residual formulation*	$\mathcal{R}_L = \{\mathrm{Id} + G \mid G \in \mathcal{G}_L\}$
Inverse residual formulation**	$\mathcal{I}_L = \{(\mathrm{Id} + G)^{-1} \mid G \in \mathcal{G}_L\}$
Monotone formulation	$\mathcal{M}_L = \left\{\left(\frac{\mathrm{Id}+G}{2}\right)^{-1} - \mathrm{Id} \,\middle|\, G \in \mathcal{G}_L\right\}$

Theorem 4. *For* $0 \leq L < 1$,

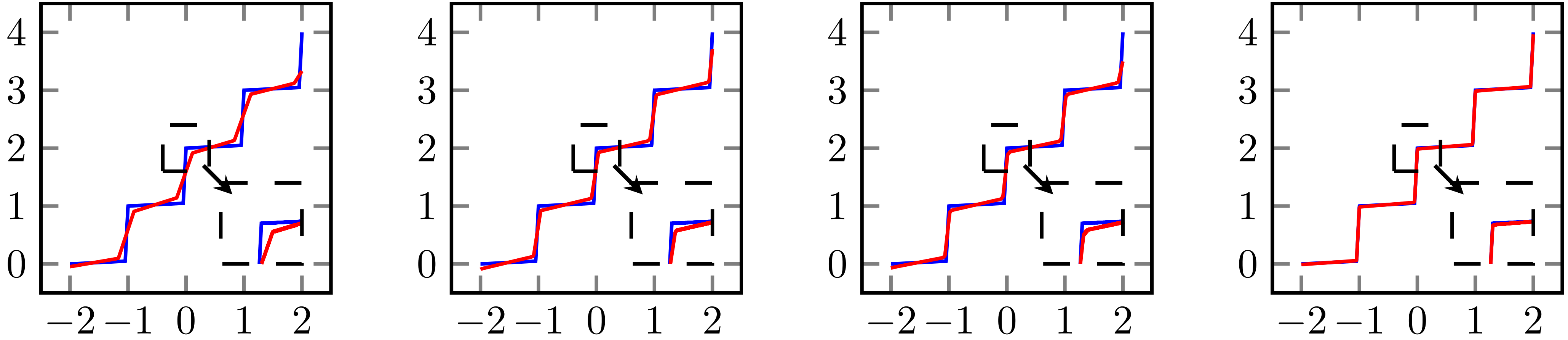
(i) $\mathcal{I}_L = \frac{1}{1-L^2}\mathcal{R}_L$, *(ii)* $\mathcal{M}_L = \frac{1+L^2}{1-L^2}\mathcal{R}_{\frac{2L}{1+L^2}}$, *(iii)* $\mathcal{R}_L \subsetneq \mathcal{M}_L$, *(iv)* $\mathcal{I}_L \subsetneq \mathcal{M}_L$.



* Behrmann et al., Invertible Residual Networks, ICML 2019.
Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.
** Lu et al., Implicit Normalizing Flows, ICLR 2021. Note that in their work one implicit block is defined as the composition of one residual formulation and one inverse residual formulation; we analyze a smaller unit without loss of generality.

Experimental Results

- 1-D fitting experiment



(a) $\mathcal{R}_L \circ \mathcal{R}_L$ (MSE: 1.6×10^{-2})
(b) $\mathbb{R}^+\mathcal{R}_L \circ \mathbb{R}^+\mathcal{R}_L$ (MSE: 5.8×10^{-3})
(c) $\mathbb{R}^+\mathcal{I}_L \circ \mathbb{R}^+\mathcal{R}_L$ (MSE: 4.5×10^{-3})
(d) $\mathbb{R}^+\mathcal{M}_L \circ \mathbb{R}^+\mathcal{M}_L$ (MSE: 7.6×10^{-5})

Figure 4: Comparison of $\mathcal{R}_L \circ \mathcal{R}_L$, $\mathbb{R}^+\mathcal{R}_L \circ \mathbb{R}^+\mathcal{R}_L$, $\mathbb{R}^+\mathcal{I}_L \circ \mathbb{R}^+\mathcal{R}_L$, and $\mathbb{R}^+\mathcal{M}_L \circ \mathbb{R}^+\mathcal{M}_L$. All experiments except (a) are performed with learnable scaling (multiplying by $\mathbb{R}^+$). Blue and red lines represent the target function and the approximation by neural networks, respectively.

Experimental Results

- 2-D toy density modelling experiments

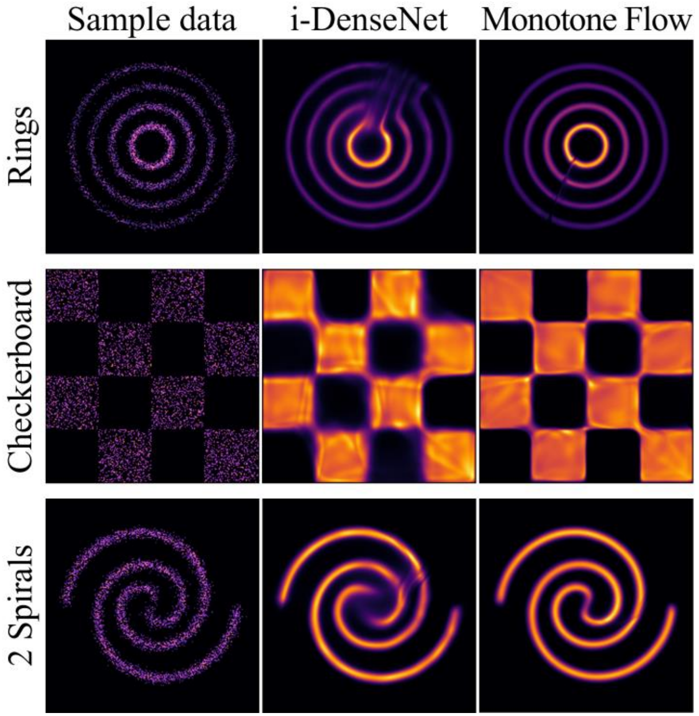


Figure 5: 2D toy density modeling results (full results in Appendix D).

Table 1: Toy density modelling results in nats. We display the average of the test loss for the last 20 tests at checkpoints (iterations 48100, 48200, ..., 50000) for a single run.

Data	i-DenseNet	Monotone Flow
2 Spirals	2.729	**2.658**
8 Gaussians	**2.840**	**2.840**
Checkerboard	3.609	**3.540**
Circles	3.280	**3.276**
Moons	2.401	**2.400**
Pinwheel	2.343	**2.333**
Rings	2.884	**2.665**
Swissroll	2.680	**2.676**

Experimental Results

- Image density modelling experiments

Table 2: Density estimation results on images in bits-per-dimension (bpd) with the number of parameters of each model. All numbers except for the last row are with uniform dequantization. VDQ: variational dequantization.

	MNIST		CIFAR-10		ImageNet32		ImageNet64	
Model	bpd ↓	params	bpd ↓	params	bpd ↓	params	bpd ↓	params
Real NVP [3]	1.06	-	3.49	6.4M	4.28	46.0M	3.98	96.0M
Glow [4]	1.05	-	3.35	44.2M	4.09	66.1M	3.81	111.1M
FFJORD [36]	0.99	-	3.40	-	-	-	-	-
i-ResNet [5]	1.06	-	3.45	44.2M	-	-	-	-
Residual Flow [6]	0.97	16.6M	3.28	25.2M	4.01	47.1M	3.76	53.3M
i-DenseNet [7]	-	-	3.25	24.9M	3.98	47.0M	-	-
Monotone Flow	**0.928**	20.9M	**3.215**	24.9M	**3.961**	47.0M	**3.734**	48.9M
Monotone Flow + VDQ	-	-	**3.062**	46.9M	**3.901**	69.0M	-	-


(a) CIFAR-10 train data.


(b) Monotone Flows trained on CIFAR-10.


(c) ImageNet32 train data.


(d) Monotone Flows trained on ImageNet32.

Figure 6: Train data and generated samples of CIFAR-10 and ImageNet32.

Summary

- A normalizing flow based on monotone operators – architecturally flexible while bypassing the Lipschitz constraint.
- A new activation function Concatenated Pila to improve gradient flow.
- Theoretical analysis shows monotone formulation is strictly more expressive than baselines. [*, **]
- On experiments, Monotone Flows consistently outperform comparable baselines on toy datasets and multiple image density estimation benchmarks.

* Behrmann et al., Invertible Residual Networks, ICML 2019.
  Chen et al., Residual Flows for Invertible Generative Modeling, NeurIPS 2019.

** Lu et al., Implicit Normalizing Flows, ICLR 2021.

Thanks for listening!