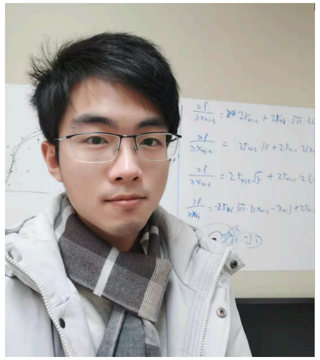
Implicit Normalizing Flows

ICLR 2021 spotlight

Cheng Lu, Jianfei Chen, Chongxuan Li, Qiuhao Wang, Jun Zhu*



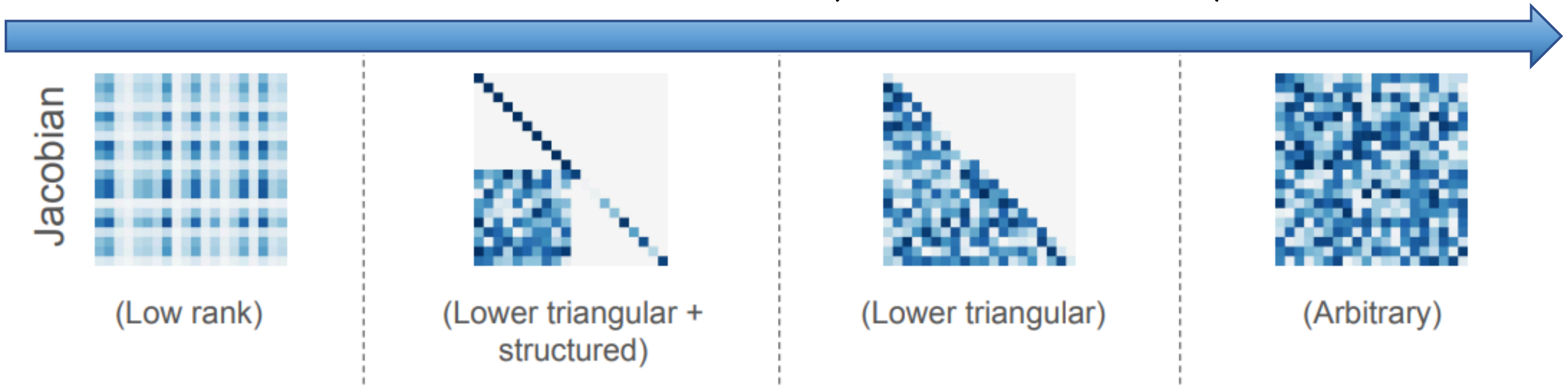

Requirements for Normalizing Flows

$$\log p_x(x) = \log p_z(f(x)) + \log\left|\det\left(\frac{df(x)}{dx}\right)\right|$$

1. An invertible function 2. Tractable log-determinant of the Jacobian

A crucial problem: Finding **rich** model families that have **tractable** log-determinants.

richer function families (Jacobian structure)



Planar NF, Sylvester NF	NICE, Real NVP, Glow	IAF, MAF, NAF	i-ResNet, ResFlow, NeuralODE, FFJORD

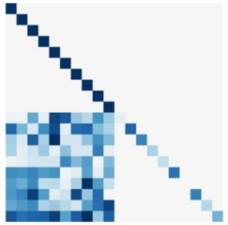
Figure from: http://www.cs.toronto.edu/~rtqichen/pdfs/residual_flows_slides.pdf

Lipschitz constraints of Jacobians

Free-form is not enough

Additive Coupling

$$\mathbf{y} = f(\mathbf{x})$$
$$\mathbf{y}_1 = \mathbf{x}_1$$
$$\mathbf{y}_2 = \mathbf{x}_2 + t(\mathbf{x}_1)$$



Structured Jacobian

$$\mathrm{Lip}(f) \leq 1 + \mathrm{Lip}(t)$$

Arbitrary Lipschitz

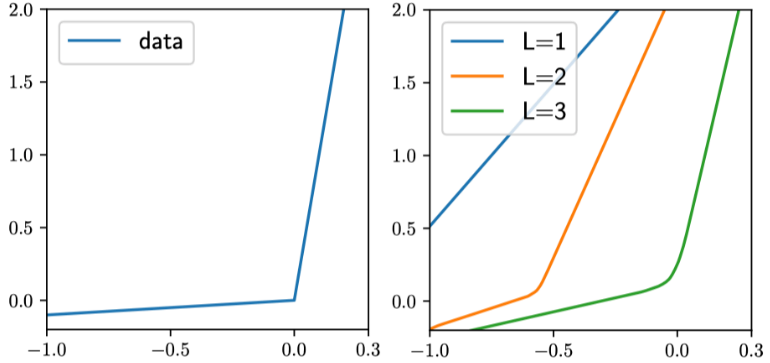
Residual Flows

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$$

Free-form Jacobian

$$\mathrm{Lip}(f) \leq 1 + \mathrm{Lip}(g) < 2$$

Bounded Lipschitz



(a) Target function (b) ResFlow

A 1-D function fitting example.

Due to the Lipschitz constraints of ResFlows, fitting a function with Lipschitz constant L needs at least $\log_2 L$ layers.

Implicit Function Theorem

Another way to define invertible mappings

Let $F\colon\ \mathbb{R}^{2d} \to \mathbb{R}^d$ be a continuously differentiable function. Let $\boldsymbol{z}$ and $\boldsymbol{x}$ be two variables in $\mathbb{R}^d$. If $\frac{\partial F(\boldsymbol{z},\boldsymbol{x})}{\partial \boldsymbol{z}}$ and $\frac{\partial F(\boldsymbol{z},\boldsymbol{x})}{\partial \boldsymbol{x}}$ are invertible matrices for any $\boldsymbol{z},\boldsymbol{x} \in \mathbb{R}^d$, then

$F(\boldsymbol{z},\boldsymbol{x}) = 0$ defines a unique and invertible mapping

$f\colon\ \mathbb{R}^d \to \mathbb{R}^d$ such that $\boldsymbol{z} = f(\boldsymbol{x})$

A special case: for any previous normalizing flow model $f(\boldsymbol{x}) = \boldsymbol{z}$, define

$$F(\boldsymbol{z},\boldsymbol{x}) = f(\boldsymbol{x}) - \boldsymbol{z}$$

Explicit invertible functions are special cases of implicit functions

Implicit Normalizing Flow (ImpFlow)

Defined by a specific form of implicit functions

Let $g_{\boldsymbol{z}}: \mathbb{R}^d \to \mathbb{R}^d,\ \mathrm{Lip}(g_{\boldsymbol{z}}) < 1,\ g_{\boldsymbol{x}}: \mathbb{R}^d \to \mathbb{R}^d,\ \mathrm{Lip}(g_{\boldsymbol{x}}) < 1,$ define

$$F(\boldsymbol{z}, \boldsymbol{x}) = g_{\boldsymbol{x}}(\boldsymbol{x}) - g_{\boldsymbol{z}}(\boldsymbol{z}) + \boldsymbol{x} - \boldsymbol{z}$$

Then $F(\boldsymbol{z}, \boldsymbol{x}) = 0$ defines a **unique** and **invertible** mapping $f: \mathbb{R}^d \to \mathbb{R}^d$ such that $\boldsymbol{z} = f(\boldsymbol{x})$.

Identity mapping

A quick check: $(g_{\boldsymbol{x}} + \mathrm{Id})(\boldsymbol{x}) = g_{\boldsymbol{x}}(\boldsymbol{x}) + \boldsymbol{x} = g_{\boldsymbol{z}}(\boldsymbol{z}) + \boldsymbol{z} = (g_{\boldsymbol{z}} + \mathrm{Id})(\boldsymbol{z})$

$$\Rightarrow \quad z = ((g_z + \mathrm{Id})^{-1}(g_x + \mathrm{Id}))(x)$$

Inverse ResFlow (implicit)

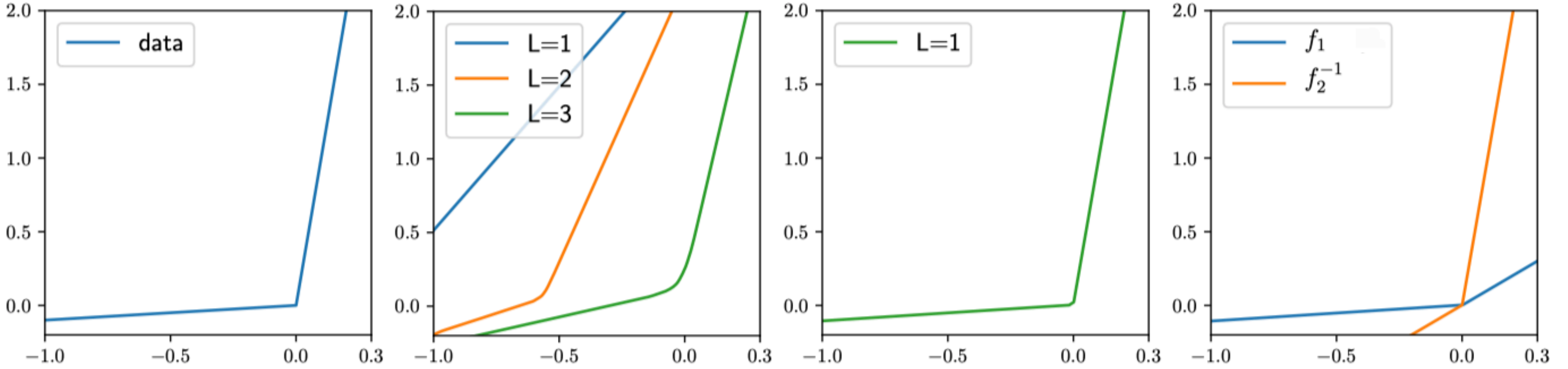
ResFlow (explicit)

An Impflow is a composition of a forward ResFlow and another inverse ResFlow.

Relaxing the Lipschitz Constraints

Also have free-form Jacobians

1-D example



(a) Target function (b) ResFlow (c) ImpFlow (d) Composition

Intuitively, the forward ResFlow handles small Lipschitz parts, and the inverse ResFlow handles large Lipschitz parts.

$$z = ((g_z + Id)^{-1}(g_x + Id))(x)$$

f_2^{-1} f_1

Main Theoretical Results

Function family of ImpFlows

Let $\mathcal{D}$ be the set of all bi-Lipschitz C^1 invertible functions from $\mathbb{R}^d$ to $\mathbb{R}^d$. Define

$$\mathcal{F} \coloneqq \{f \in \mathcal{D} : \inf_{\mathbf{x}\in\mathbb{R}^d, \mathbf{v}\in\mathbb{R}^d, \|\mathbf{v}\|_2=1} \mathbf{v}^T J_f(\mathbf{x}) \mathbf{v} > 0\}$$

"monotonically increasing functions"

- 1-D case: $\mathcal{F} = \{f \in C^1(\mathbb{R}) : \inf_{x\in\mathbb{R}} f'(x) > 0\}$.
- Symmetric case: gradients of strongly convex functions.

Includes "Convex Potential Flows" (ICLR 2021) [1]

Theorem $\mathcal{I} = \mathcal{F}_2 \coloneqq \{f : f = f_2 \circ f_1 \text{ for some } f_1, f_2 \in \mathcal{F}\}.$

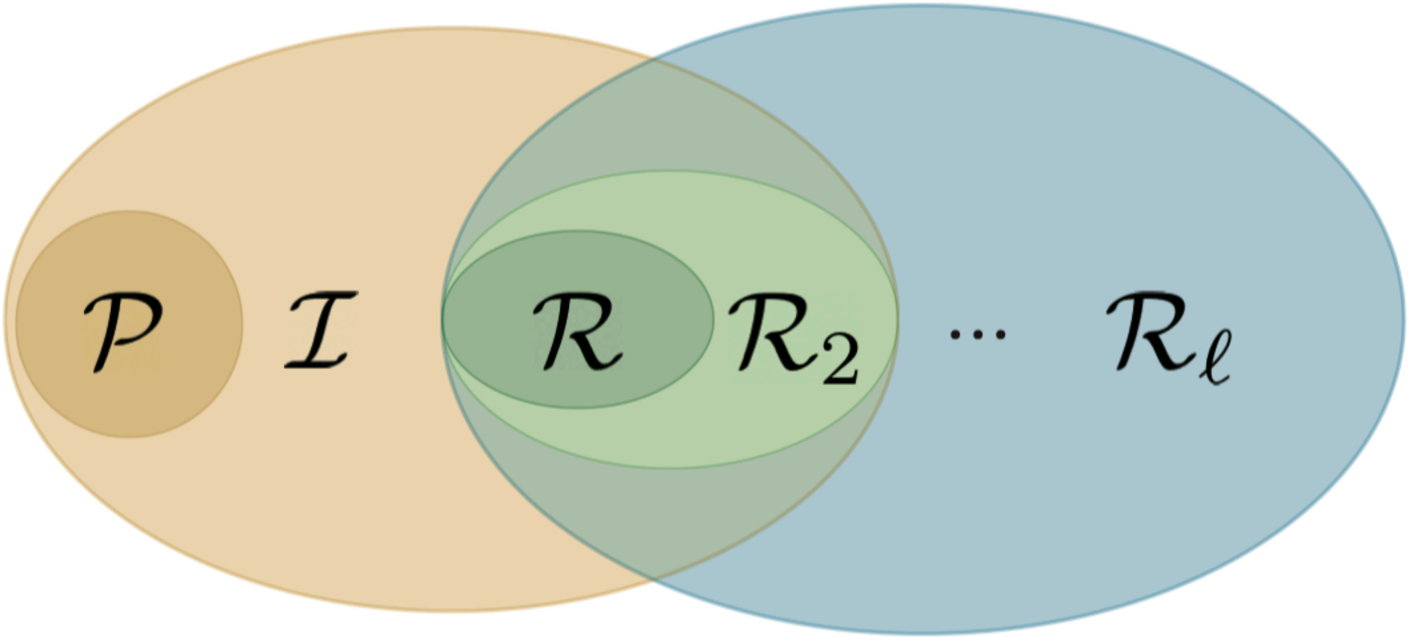
⇓

Function family of single-layer ImpFlows

[1] Huang, Chin-Wei, et al. "Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization." *arXiv preprint arXiv:2012.05942* (2020).

Main Theoretical Results

Relationship between 1-layer ImpFlow and ℓ-layer ResFlow



1. $\mathcal{R} \subsetneqq \mathcal{R}_2 \subsetneqq \mathcal{I}$.

   2-layer ResFlows are strictly included in 1-layer ImpFlows.

2. $\forall\, \ell > 0,\ \mathcal{I} \not\subseteq \mathcal{R}_\ell$.

   Any multi-layer ResFlows cannot include 1-layer ImpFlows.

Generative Modeling

Remains tractability

- Forward & Inverse: finding fixed point.

$$z = ((g_z + Id)^{-1}(g_x + Id))(x)$$

1. Simple fixed point iterations.
2. Quasi-Newton methods (i.e. Broyden's method).

- Log-determinant of Jacobians: the same unbiased estimator as ResFlows.

$$\ln p(\mathbf{x}) = \ln p(\mathbf{z}) + \ln\det(I + J_{g_{\mathbf{x}}}(\mathbf{x})) - \ln\det(I + J_{g_{\mathbf{z}}}(\mathbf{z}))$$

- Backpropagation for the fixed point: take total derivative of $F(\boldsymbol{z}, \boldsymbol{x}) = 0$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial (\cdot)} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}} J_G^{-1}(\mathbf{z}) \frac{\partial F(\mathbf{z}, \mathbf{x}; \theta)}{\partial (\cdot)}, \text{ where } G(\mathbf{z}; \theta) = g_{\mathbf{z}}(\mathbf{z}; \theta) + \mathbf{z}.$$

Model Expressiveness
Classification tasks

Table 1: Classification error rate (%) on test set of vanilla ResNet, ResFlow and ImpFlow of ResNet-18 architecture, with varying Lipschitz coefficients c.

		Vanilla	$c = 0.99$	$c = 0.9$	$c = 0.8$	$c = 0.7$	$c = 0.6$
CIFAR10	ResFlow	6.61(±0.02)	8.24 (±0.03)	8.39 (±0.01)	8.69 (±0.03)	9.25 (±0.02)	9.94 (±0.02)
	ImpFlow		**7.29** (±0.03)	**7.41** (±0.03)	**7.94** (±0.06)	**8.44** (±0.04)	**9.22** (±0.02)
CIFAR100	ResFlow	27.83(±0.03)	31.02 (±0.05)	31.88 (±0.02)	32.21 (±0.03)	33.58 (±0.02)	34.48 (±0.03)
	ImpFlow		**29.06** (±0.03)	**30.47** (±0.03)	**31.40** (±0.03)	**32.64** (±0.01)	**34.17** (±0.02)

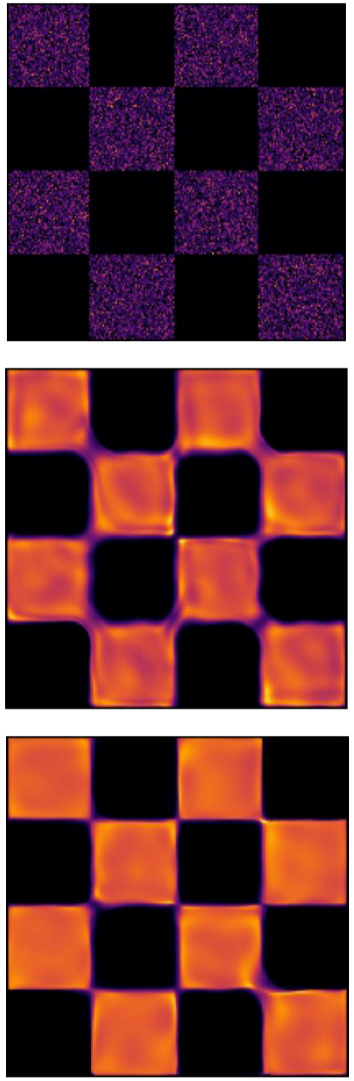
Density Estimation Experiments

Table 2: Average test log-likelihood (in nats) of tabular datasets. Higher is better.

	POWER	GAS	HEPMASS	MINIBOONE	BSDS300
RealNVP (Dinh et al., 2017)	0.17	8.33	-18.71	-13.55	153.28
FFJORD (Grathwohl et al., 2019)	0.46	8.59	-14.92	-10.43	157.40
MAF (Papamakarios et al., 2017)	0.24	10.08	-17.70	-11.75	155.69
NAF (Huang et al., 2018)	**0.62**	11.96	-15.09	**-8.86**	**157.73**
ImpFlow ($L = 20$)	0.61	**12.11**	**-13.95**	-13.32	155.68
ResFlow ($L = 10$)	0.26	6.20	-18.91	-21.81	104.63
ImpFlow ($L = 5$)	**0.30**	**6.94**	**-18.52**	**-21.50**	**113.72**

Table 3: Average bits per dimension of ResFlow and ImpFlow on CIFAR10, with varying Lipschitz coefficients c. Lower is better.

	$c = 0.9$	$c = 0.8$	$c = 0.7$	$c = 0.6$
ResFlow ($L = 12$)	3.469($\pm$0.0004)	3.533($\pm$0.0002)	3.627($\pm$0.0004)	3.820($\pm$0.0003)
ImpFlow ($L = 6$)	**3.452**($\pm$0.0003)	**3.511**($\pm$0.0002)	**3.607**($\pm$0.0003)	**3.814**($\pm$0.0005)



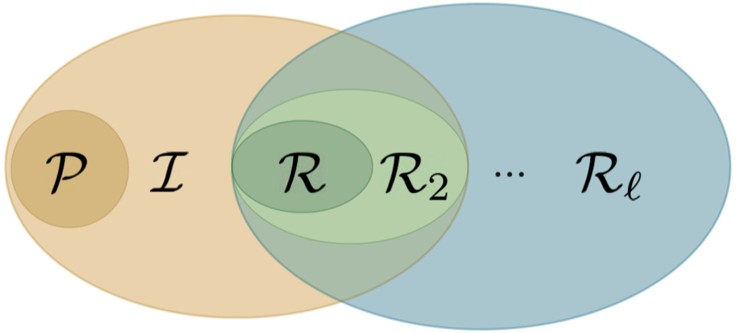
(a) Checkerboard data (5.00 bits)

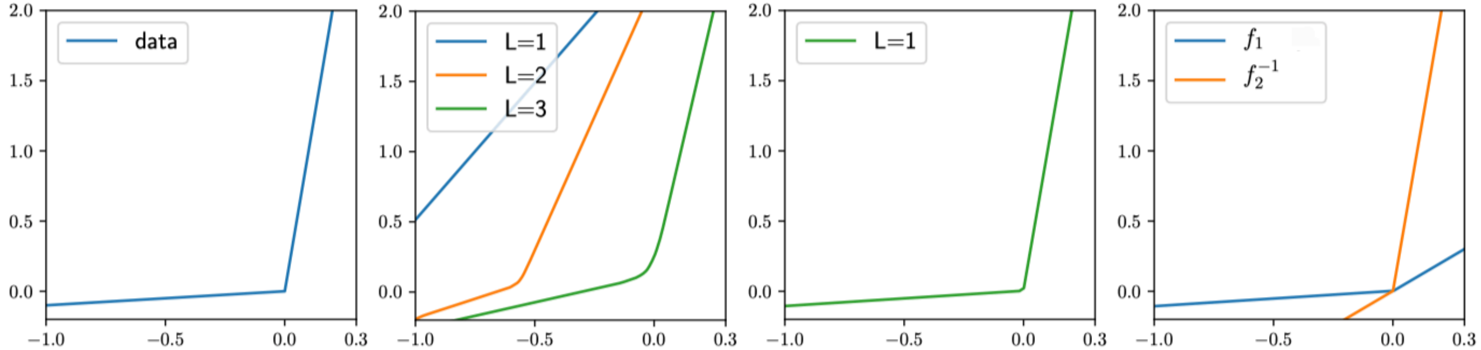
(b) ResFlow, $L = 8$ (5.08 bits)

(c) ImpFlow, $L = 4$ (5.05 bits)

Summary

- A **unique** and **invertible** mapping defined by a specific form of **implicit functions**.
- A more **powerful function space** than ResFlows, which relaxing the **Lipschitz constraints** of ResFlows.
- A **scalable** algorithm for generative modeling.





(a) Target function (b) ResFlow (c) ImpFlow (d) Composition

Thanks for Listening!

- ICLR 2021 poster ID: 3381
- Released Code: https://github.com/thu-ml/implicit-normalizing-flows
- Contact: lucheng.lc15@gmail.com