Robust model training and generalisation with Studentising flows

Simon Alexanderson   Gustav Eje Henter

{simonal,ghe}@kth.se

Division of Speech, Music and Hearing (TMH),
School of Electrical Engineering and Computer Science (EECS),
KTH Royal Institute of Technology, Stockholm, Sweden

2020-07-11

One-slide summary

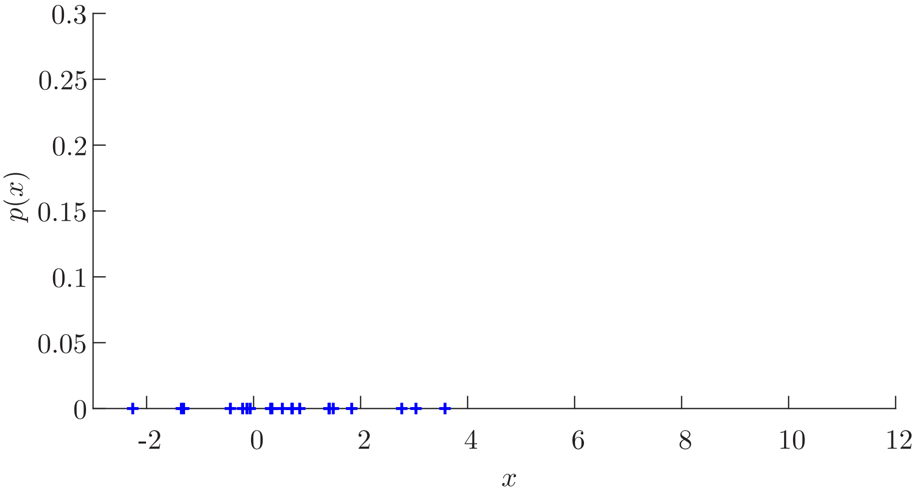
- We propose replacing Gaussian base distributions $\boldsymbol{Z}$ in normalising flows with **multivariate Student's t-distributions**
  - *Studentising flows*
- Our proposal is motivated through **statistical robustness**
- Experiments show that the proposal **stabilises training** and leads to **better generalisation**

Outline

- What is robustness?
- Robustness sits in the tails
- Tails of flow-based models
- Experimental findings

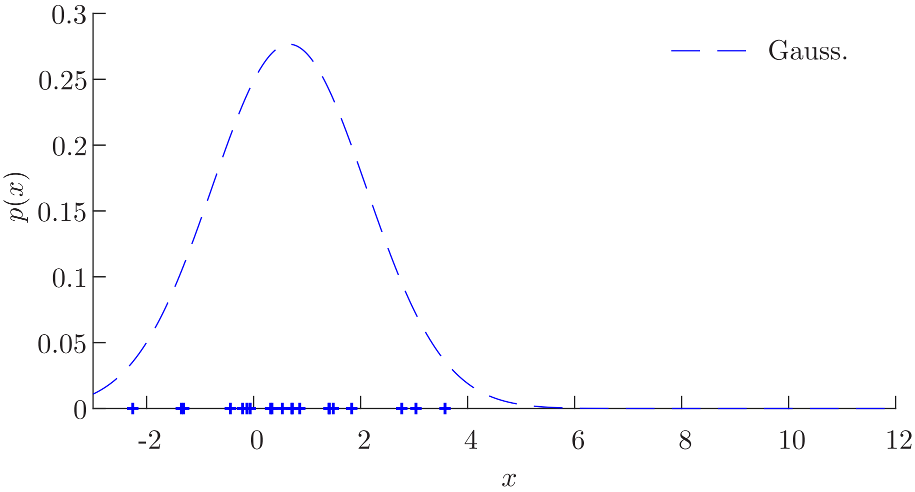
Why do we need robustness?

Generate some 1D standard normal data and fit a Gaussian:



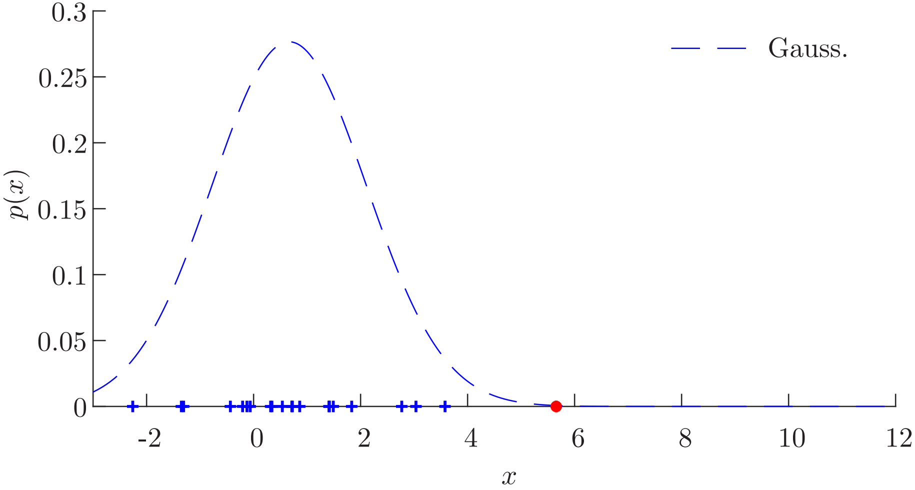
Why do we need robustness?

Generate some 1D standard normal data and fit a Gaussian:



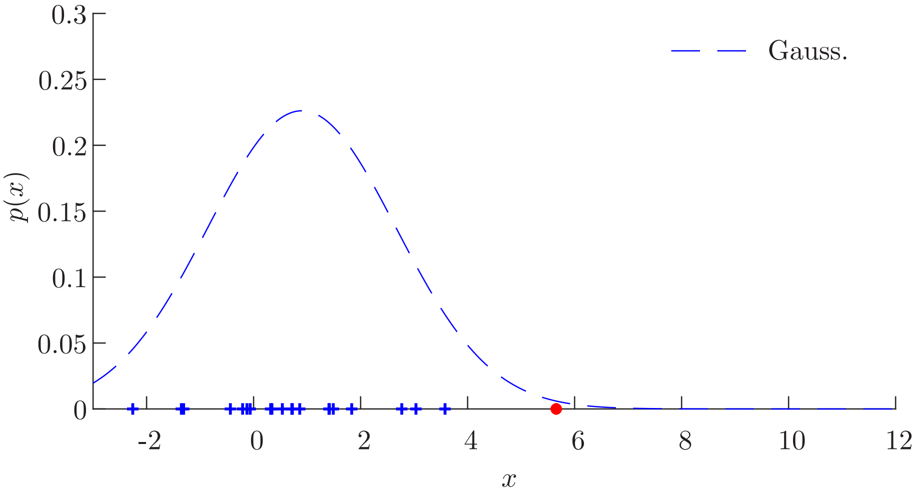
Why do we need robustness?

The fit changes if we add an outlying datapoint (red blob).



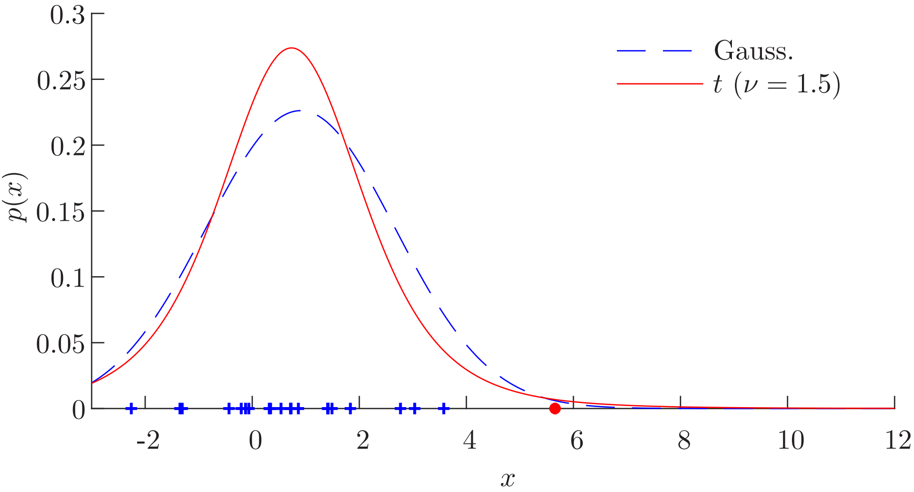
Why do we need robustness?

The fit changes if we add an outlying datapoint (red blob).



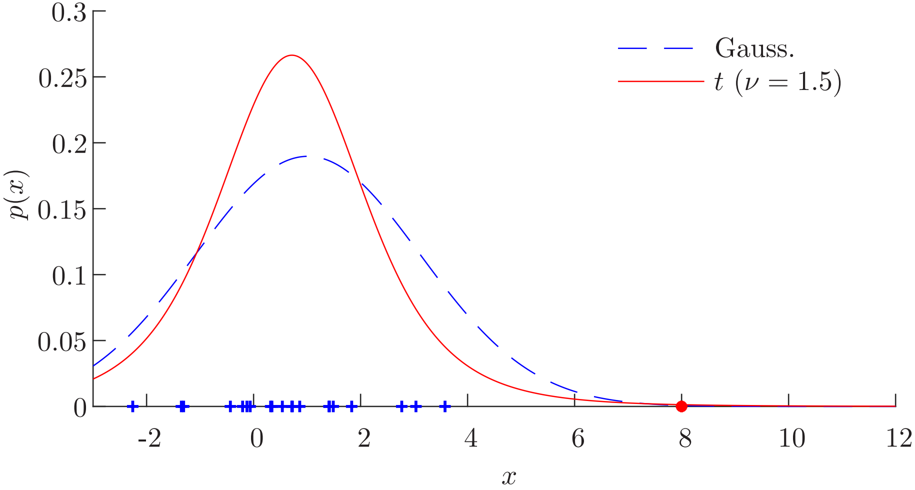
Why do we need robustness?

A fitted Student's t-distribution (red plot) is more concentrated.



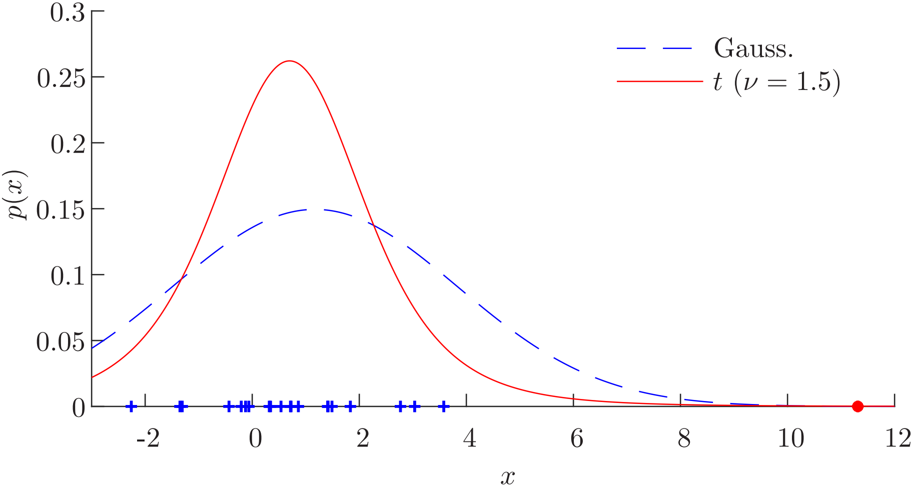
Why do we need robustness?

As the outlier is moved away, the Gaussian fit changes a lot.

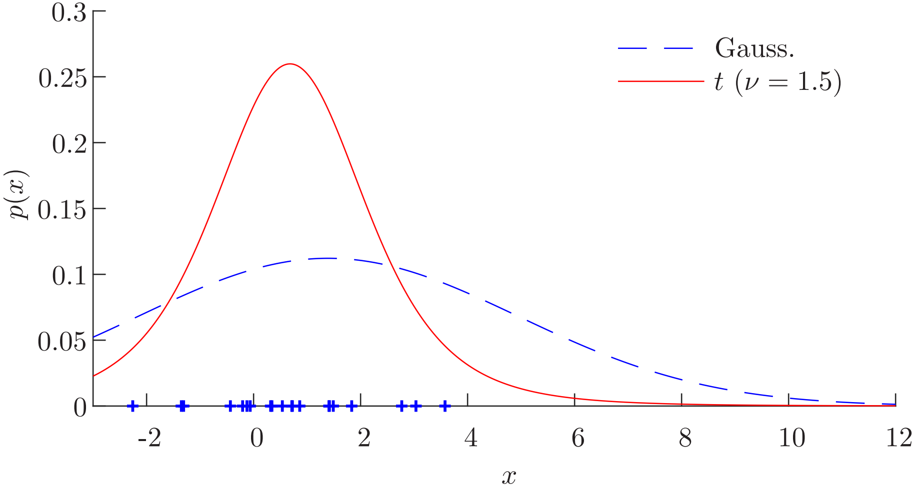


Why do we need robustness?

As the outlier is moved away, the Gaussian fit changes a lot.

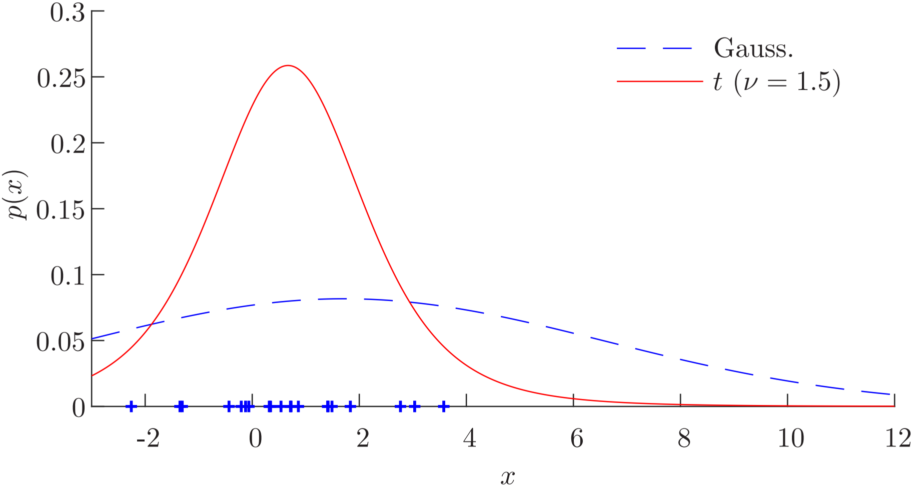


As the outlier is moved away, the Gaussian fit changes a lot.



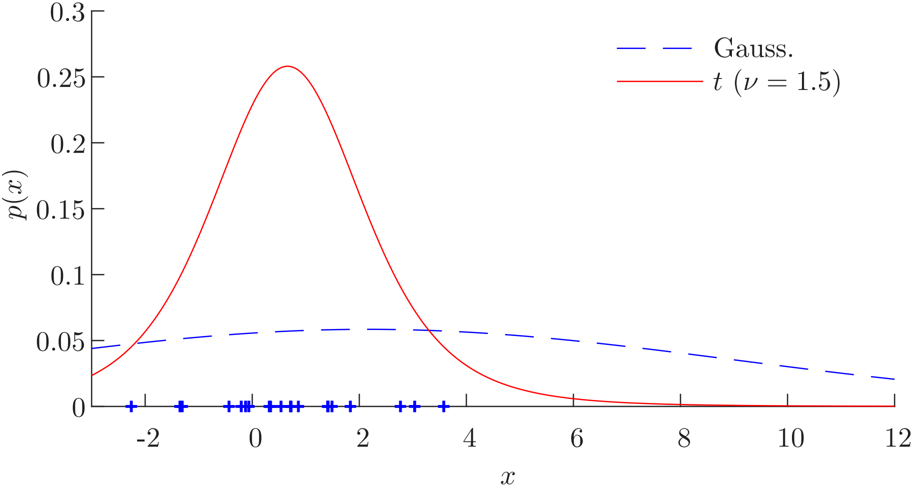
Why do we need robustness?

As the outlier is moved away, the Gaussian fit changes a lot.



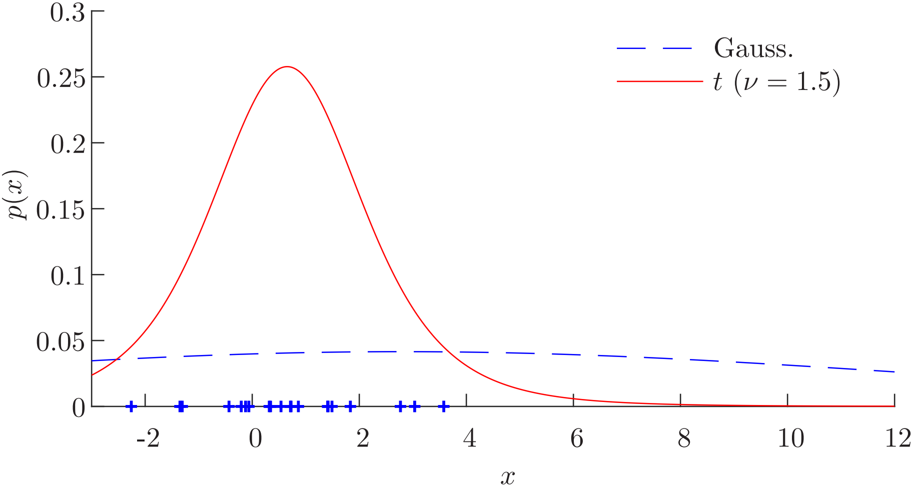
Why do we need robustness?

As the outlier is moved away, the Gaussian fit changes a lot.



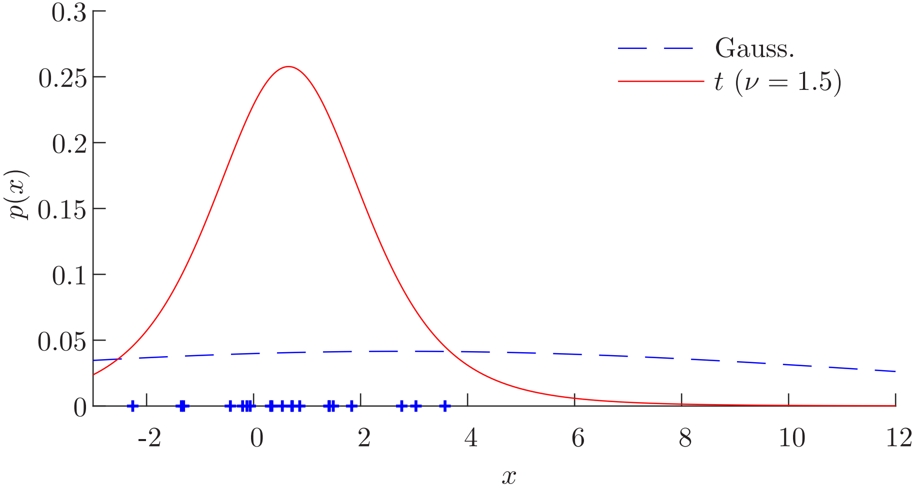
Why do we need robustness?

As the outlier is moved away, the Gaussian fit changes a lot.



Why do we need robustness?

In contrast, the Student's *t*-distribution is *statistically robust*.

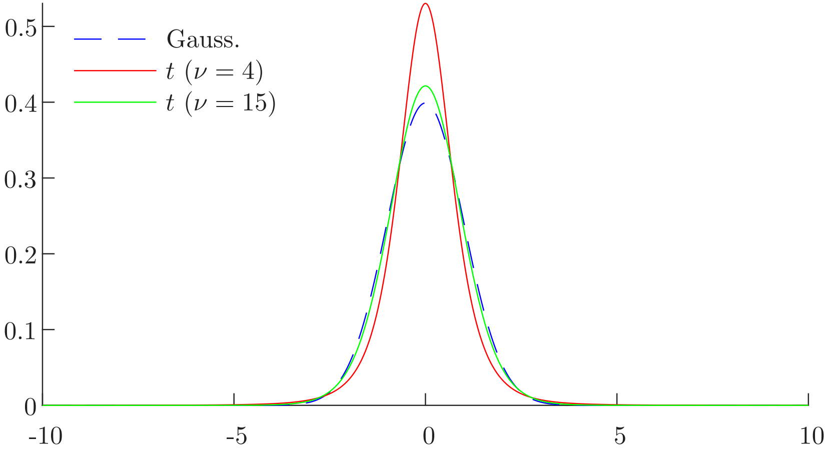


Robust statistics

Robust (*resistant*) estimator:
Adversarially corrupting a fraction η of the data ($\eta < 1/2$) only has a *bounded* effect on the estimated model parameters $\widehat{\boldsymbol{\theta}}$

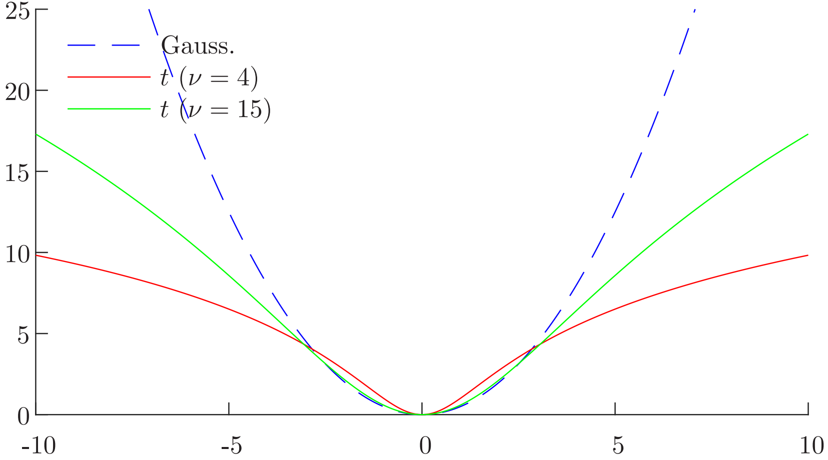
Why is Student's t robust?

The probability density functions of Gaussians and Student's t-distributions look similar.



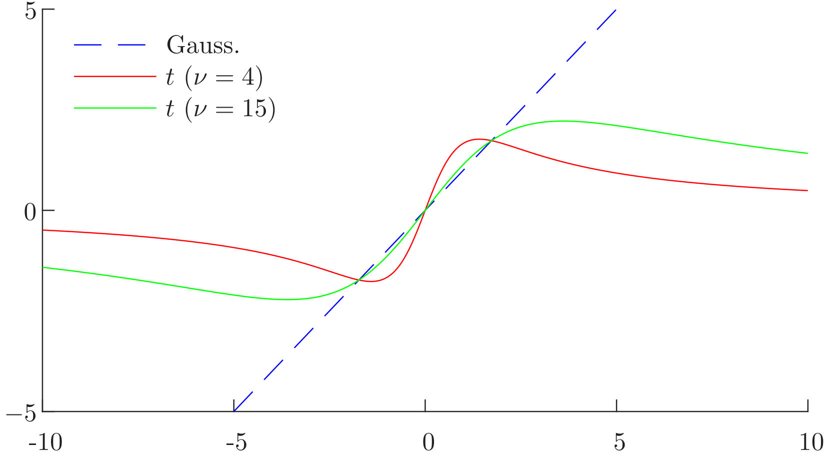
Why is Student's t robust?

The associated loss functions (the negative log-likelihood, or NLL) exhibit differences in the tails.



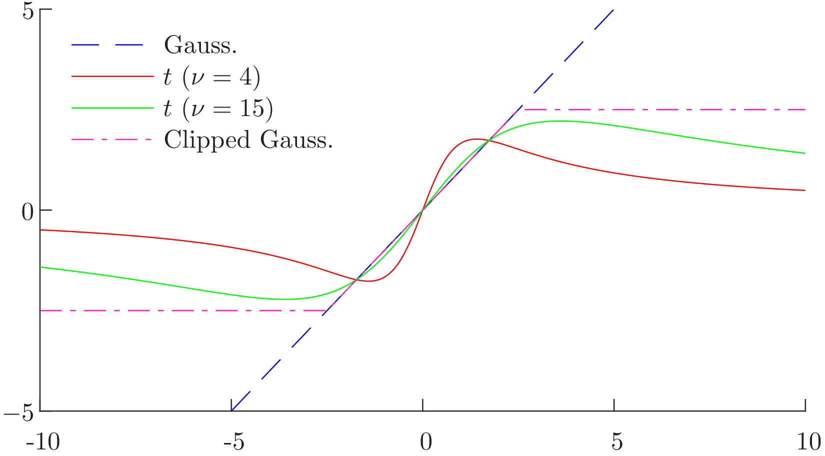
Why is Student's t robust?

The *influence function* is the gradient of the NLL. It quantifies the effect of outliers. For the t-distribution the influence function is bounded.



Why is Student's t robust?

Gradient clipping can also limit the influence of outliers, but need not converge on the maximum-likelihood model.



Related work

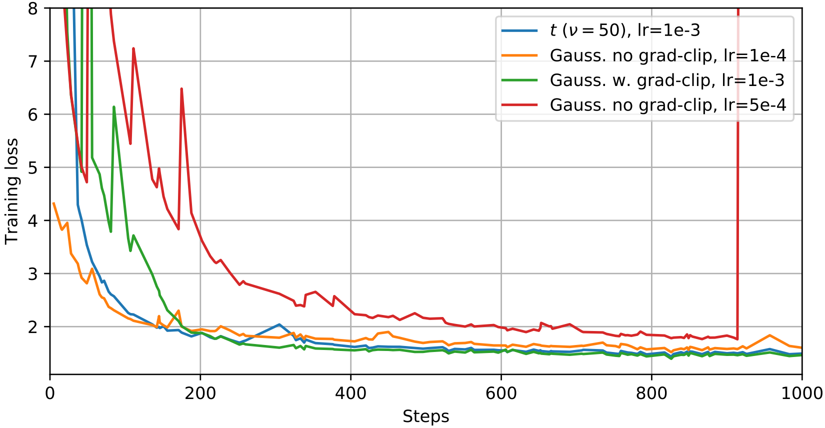
Our findings complement those in concurrent work by Jaini et al. (2020)[1]

- They show:
  - Lipschitz-continuous triangular flows $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{Z})$ with Gaussian base distributions $\boldsymbol{Z}$ cannot represent fat-tailed data
    - For example: Glow with sigmoid-transformed scale factors
  - Using multivariate t_ν-distributions allows modelling data with fat tails
- We add to this:
  - The advantages of t_ν-distributions can be understood through statistical robustness
  - Experimentally, these benefits extend to bounded data (no fat tails)

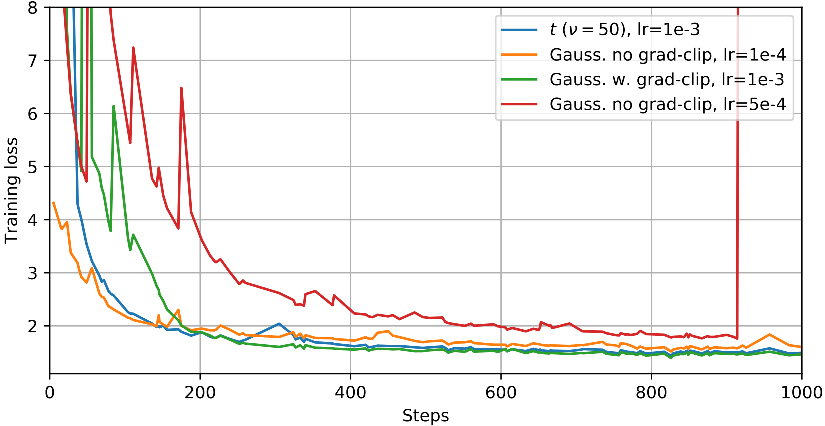
[1] Jaini, P., Kobyzev, I., Yu, Y., and Brubaker, M. Tails of Lipschitz triangular flows. In *Proc. ICML*, 2020.

Stable training

Training loss of Glow models of 64×64 CelebA data trained using Adam. The red configuration is unstable.



Reducing the learning rate (yellow), clipping gradients (green), or changing the base to a multivariate t_ν-distribution (blue) stabilises training.



Better generalisation on image data

Test set negative log-likelihood on MNIST with and without outliers from greyscale CIFAR-10. $\nu = \infty$ is the Gaussian baseline.

	Test	Clean				1% outliers			
Train	$\nu =$	∞	20	50	1000	∞	20	50	1000
Clean	NLL	1.16	1.13	1.13	1.17	1.63	1.27	1.26	1.31
	Δ	0	-0.03	-0.03	0.01	0	-0.36	-0.37	-0.32
1% outliers	NLL	1.17	1.13	1.14	1.18	1.21	1.18	1.19	1.22
	Δ	0	-0.04	-0.03	0.01	0	-0.03	-0.02	0.01

Better generalisation on image data

Test set negative log-likelihood on MNIST with and without outliers from greyscale CIFAR-10. $\nu = \infty$ is the Gaussian baseline.

Train	Test	Clean				1% outliers			
	$\nu =$	∞	20	50	1000	∞	20	50	1000
Clean	NLL	1.16	1.13	1.13	1.17	1.63	1.27	1.26	1.31
	Δ	0	-0.03	-0.03	0.01	0	-0.36	-0.37	-0.32
1% outliers	NLL	1.17	1.13	1.14	1.18	1.21	1.18	1.19	1.22
	Δ	0	-0.04	-0.03	0.01	0	-0.03	-0.02	0.01

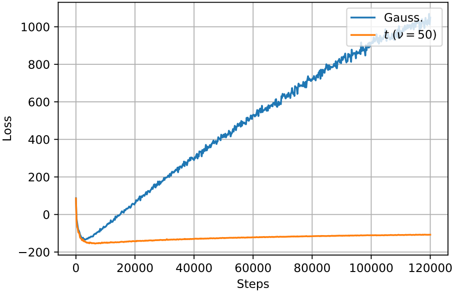
Better generalisation on image data

Test set negative log-likelihood on MNIST with and without outliers from greyscale CIFAR-10. $\nu = \infty$ is the Gaussian baseline.

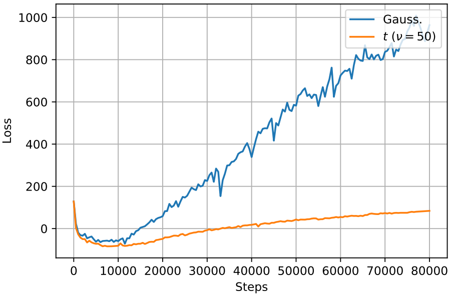
	Test	Clean				1% outliers			
Train	$\nu =$	∞	20	50	1000	∞	20	50	1000
Clean	NLL	1.16	1.13	1.13	1.17	1.63	1.27	1.26	1.31
	Δ	0	-0.03	-0.03	0.01	0	-0.36	-0.37	-0.32
1% outliers	NLL	1.17	1.13	1.14	1.18	1.21	1.18	1.19	1.22
	Δ	0	-0.04	-0.03	0.01	0	-0.03	-0.02	0.01

Better generalisation on more complex data

In probabilistic motion modelling, flow-based models are the current state of the art in terms of output quality. However, they are quite overfitted.



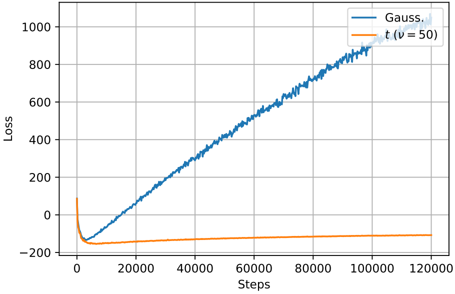
Locomotion synthesis



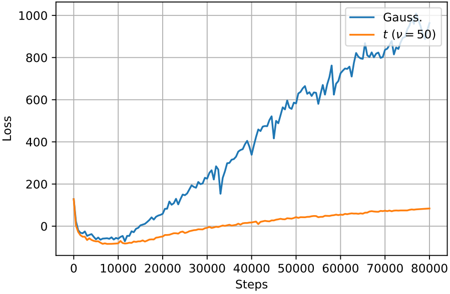
Gesture generation

Better generalisation on more complex data

Studentising flows (yellow) perform equally well on training data but greatly reduce overfitting for locomotion and gesture-modelling tasks.



Locomotion synthesis



Gesture generation

Please see our paper for more!

- Additional experiments and results
- Connections between:
  - Consistency and asymptotic efficiency
  - Statistical robustness
  - Machine-learning best practises
- Code