WP3.1 Generative Neural Networks
Phase III - Themed Meeting on Deep Learning

Queen’s University Belfast
14 November 2019

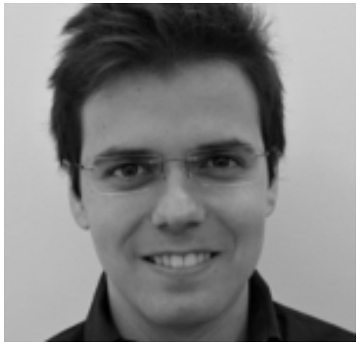
Dr Nikolaos Dionelis
Research Associate

Academics:

Prof. Sotirios Tsaftaris

Dr Mehrdad Yaghoobi

Dr Joao Mota



Dr Sen Wang

https://udrc.eng.ed.ac.uk/udrc-themed-meeting-machine-learning-and-deep-learning

Generative Neural Networks for AD

- Anomaly Detection (AD):
  - Identification of samples that differ from typical data
  - Anomaly not known in advance, before inference
- **Aim**: Detect strong anomalies
  - Strong anomalies: Near boundary

    Weak anomalies: Far from boundary
  - Specifically adversarial anomalies:

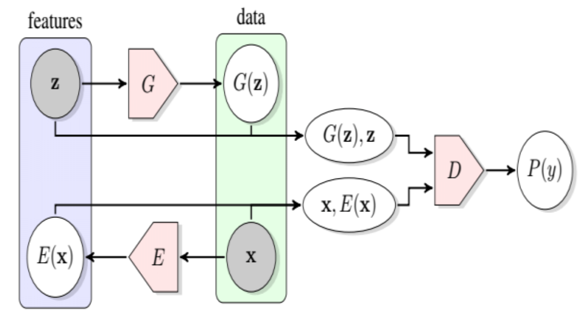
    Anomalies close to high-probability normal samples
  - Provide decision boundaries for inference of within and OoD
- Application areas:
  - X-ray contraband detection; Electro-optical sensors

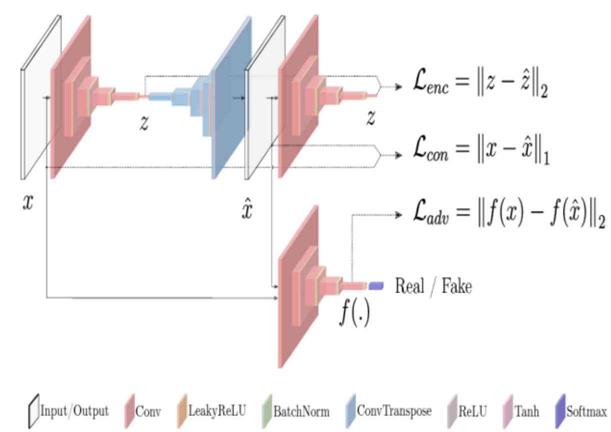
Current Methodologies for AD

- Generative Models (GMs) used for AD:
  - Generative Adversarial Networks (GANs)
  - Autoencoders (AEs)
  - Invertible GMs: Flows
- Architecture / Loss function / Algorithm
- GAN generator or discriminator for AD

- Other Invertible GMs:
  - IResNet, ResFlow: Not yet used for AD
  - Compute probability at any point in the high-dimensional image data space

EGBAD Model Architecture



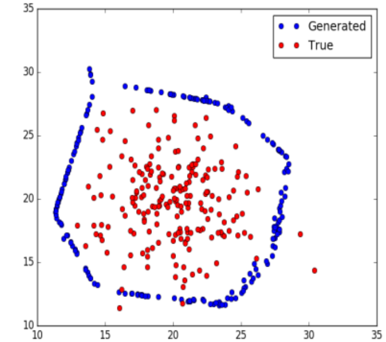


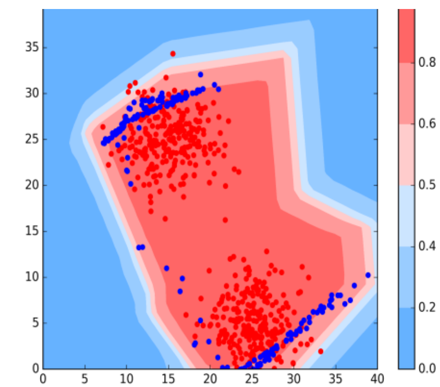
GANomaly Model Architecture

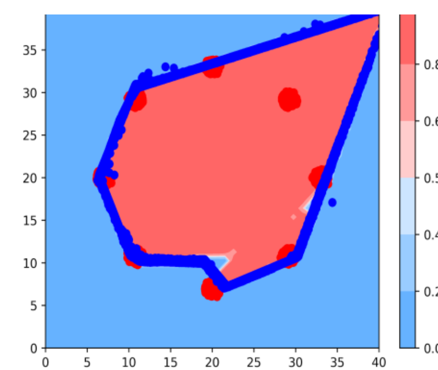
Discernible Limitations for Practical AD

- Shortcomings of current methodologies:
  - Leave-one-out evaluation
  - Lack strong anomalies definition
- Anomalies not confined to a finite labelled set
  - Broader definition of anomaly
  - Complement of the support
- Rarity problem
  - Sample the tails
  - Sampling complexity
- Multi-mode distribution estimation

- **Aim**: Address these challenges





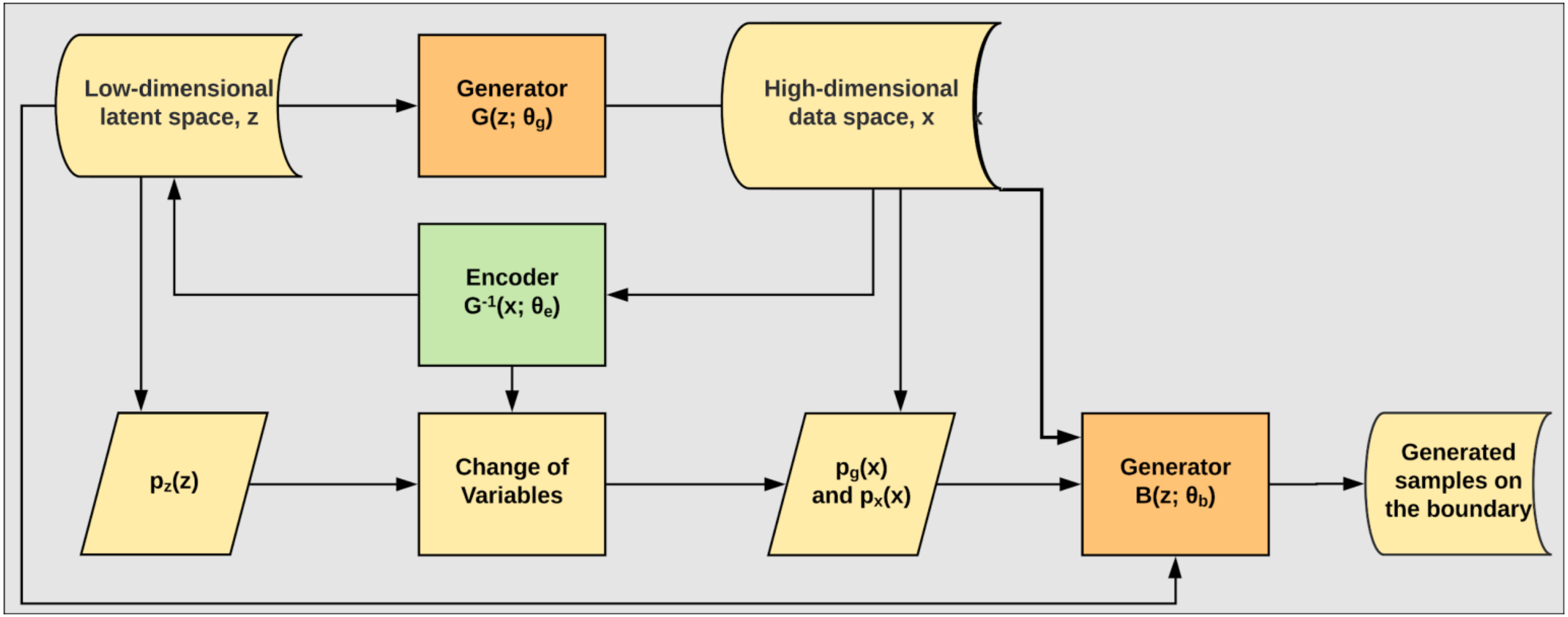


Proposed BDSGM and Contributions

- Develop the IResNet- and boundary-based model:
  Boundary of Distribution's Support Generative Model (BDSGM)

- For AD: Accurate boundary estimation is key
- Train an invertible generative model, IResNet
- Create an algorithm for sample generation on the boundary
  - Obviate the rarity and sampling complexity problems
- Improve the leave-one-out evaluation methodology
- Generation of strong anomalies
  - Specifically of adversarial anomalies
- Evaluate the use of inference for anomaly detection

Flowchart of BDSGM



- Train an invertible model to fit the normal data distribution
  - Train the IResNet to learn Generator G(**z**) and G^{-1}(**x**)
- Create and train the B(**z**) to generate samples on the boundary

Loss Function for B(z; $\boldsymbol{\theta}_b$)

- Given a data distribution, $p_{\mathbf{x}}(\mathbf{x})$: Approximate its probability density with an IResNet, $G(\mathbf{z})$, to obtain $p_g(\mathbf{x})$.
- $B(\mathbf{z})$ = Mapping from latent space, $\mathbf{z}$, to image data space, $\mathbf{x}$
- $\boldsymbol{\theta}_b$= Parameters of $B(\mathbf{z})$
- Minimize proposed loss function L:

$$\operatorname{argmin}_{\boldsymbol{\theta}_b} L(\boldsymbol{\theta}_b, \mathbf{z}, \mathbf{x}, G, \lambda_1, \lambda_2)$$

$$L(\boldsymbol{\theta}_b, \mathbf{z}, \mathbf{x}, G, \lambda_1, \lambda_2) = L_0(\boldsymbol{\theta}_b, \mathbf{z}, G) + \lambda_1 L_1(\boldsymbol{\theta}_b, \mathbf{z}, \mathbf{x}) + \lambda_2 L_2(\boldsymbol{\theta}_b, \mathbf{z})$$

- Run the Stochastic Gradient Descent (SGD) algorithm on the proposed loss function to obtain $\boldsymbol{\theta}_b$

Three Terms of Proposed Loss

- L_0: Penalize probability density to find the boundary
- L_1: Distance from a point to a set
  - Penalize distance from normality
- L_2: Scattering, dispersion, and diversity
  - Avoid mode collapse

$$L(\boldsymbol{\theta}_b, \mathbf{z}, \mathbf{x}, G, \lambda_1, \lambda_2) = L_0(\boldsymbol{\theta}_b, \mathbf{z}, G) + \lambda_1 L_1(\boldsymbol{\theta}_b, \mathbf{z}, \mathbf{x}) + \lambda_2 L_2(\boldsymbol{\theta}_b, \mathbf{z})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left[p_g(B(\mathbf{z}_i; \boldsymbol{\theta}_b)) + \lambda_1 \min_{j=1}^{M} ||B(\mathbf{z}_i; \boldsymbol{\theta}_b) - \mathbf{x}_j||_2 \; + \lambda_2 \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \frac{||\mathbf{z}_i - \mathbf{z}_j||_2}{||B(\mathbf{z}_i; \boldsymbol{\theta}_b) - B(\mathbf{z}_j; \boldsymbol{\theta}_b)||_2} \right]$$

- N = Batch size
- M = Sample size

Expansion of First Term

- Use change of variables formula:

$$L_0(\boldsymbol{\theta}_b, \mathbf{z}, G) = \frac{1}{N}\sum_{i=1}^{N} p_g(B(\mathbf{z}_i; \boldsymbol{\theta}_b))$$

$$= \frac{1}{N}\sum_{i=1}^{N} \left[p_{\mathbf{z}}(G^{-1}(B(\mathbf{z}_i; \boldsymbol{\theta}_b)))\, |\det \mathbf{J}_G(B(\mathbf{z}_i; \boldsymbol{\theta}_b))|^{-1}\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N} \left[\exp(\log(p_{\mathbf{z}}(G^{-1}(B(\mathbf{z}_i; \boldsymbol{\theta}_b)))) - \log(|\det \mathbf{J}_G(B(\mathbf{z}_i; \boldsymbol{\theta}_b))|))\right]$$

- Depends on: B(**z**), G^{-1}(**x**), det $\mathbf{J}_G$(**x**), $p_{\mathbf{z}}$(**z**)
- Standard Gaussian distribution, **z** ~ N(**0**; **I**)
- Inference: Queried test sample, **x***
  - Anomaly if $p_g(\mathbf{x}^*) = \exp(\log(p_{\mathbf{z}}(G^{-1}(\mathbf{x}^*))) - \log(|\det \mathbf{J}_G(\mathbf{x}^*)|)) < \varepsilon$
  - Normal otherwise

Experimental Setup of Proposed Model

- PyTorch vectorized implementation of the BDSGM
- Synthetic data
  - Two-dimensional uni- and multi-mode Gaussian distributions
- Closed-Form Solution (CFS) evaluation of the first term:

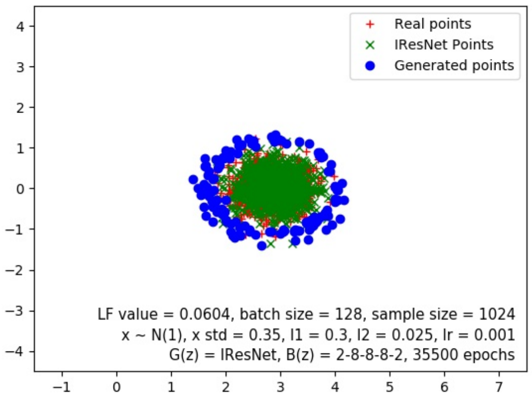
$$L_0(\boldsymbol{\theta}_b, \mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N}\Big[\left((2\pi)^d \det \boldsymbol{\Sigma}\right)^{-0.5} \times \exp\left(-0.5\,(B(\mathbf{z}_i;\boldsymbol{\theta}_b) - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (B(\mathbf{z}_i;\boldsymbol{\theta}_b) - \boldsymbol{\mu})\right)\Big]$$

- Train the CFS-based BDSGM
  - Generator $B(\mathbf{z};\, \boldsymbol{\theta}_b)$ network architecture
  - Hyper-parameters: $\lambda_1 = 0.3$ and $\lambda_2 = 0.025$
  - Sample size M = 1024 and batch size N = 256
- Train the IResNet-based BDSGM

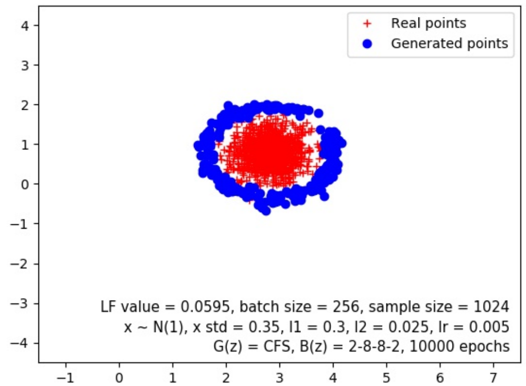
Uni-Mode Evaluation of BDSGM

- Boundary formation of IResNet-based BDSGM
- Compare: Outputs, loss function (LF) values, convergence rate

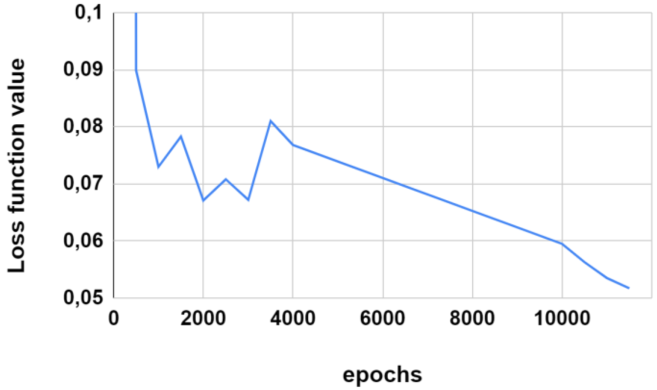
(a) IResNet- BDSGM; B(z) FC 2-8-8-8-2



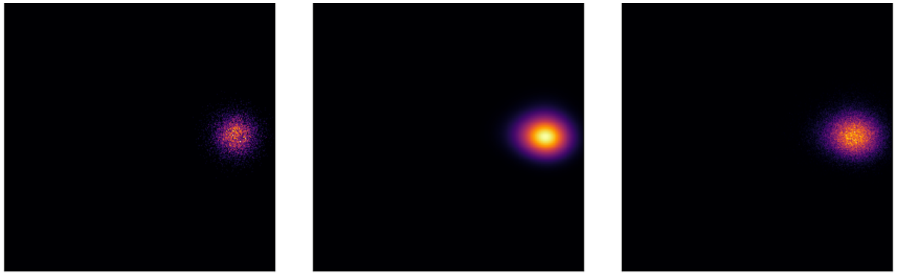
(b) CFS- BDSGM



(c) Learning curve, LF values

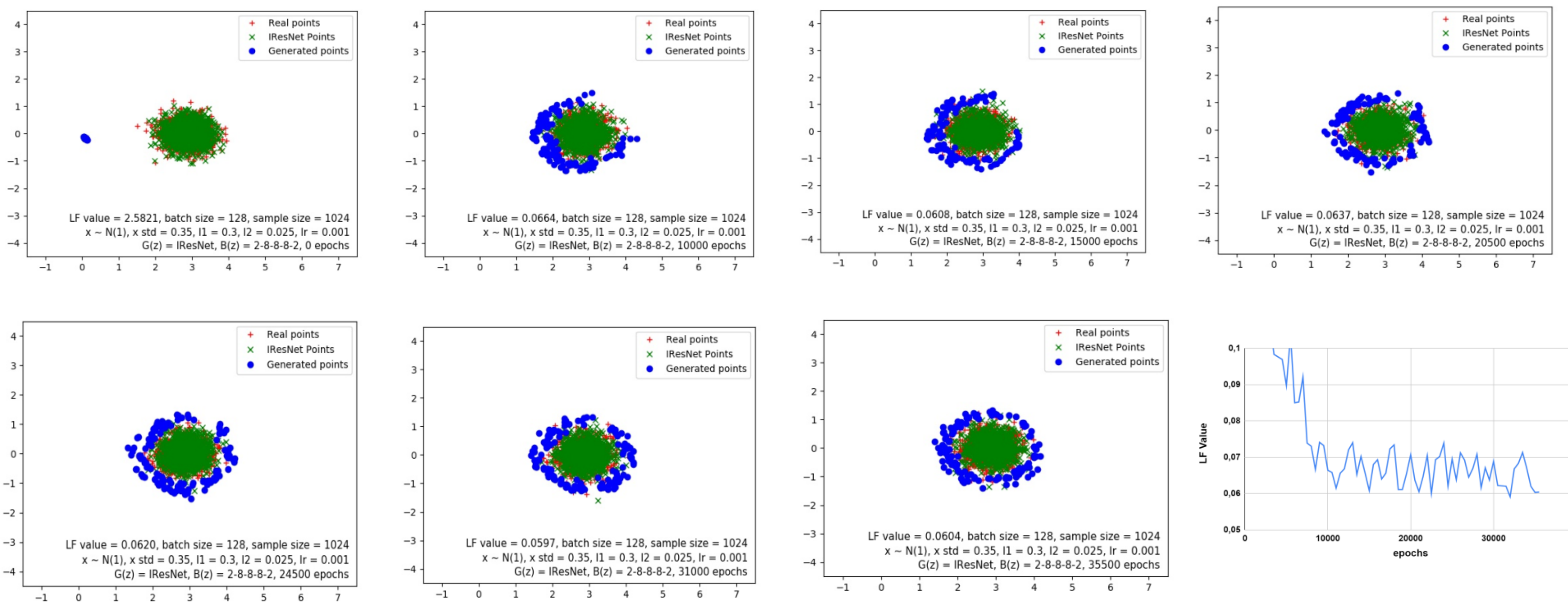


(d) IResNet: Input samples (left), output probability density (middle), and output samples (right)



Training Evolution

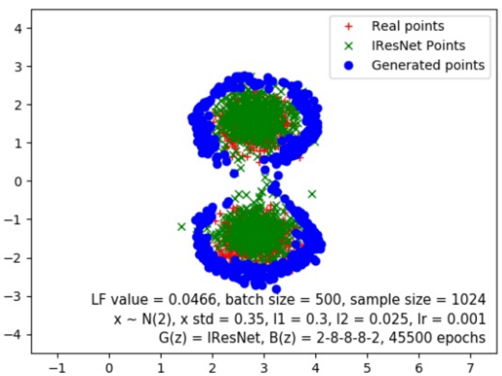
- BDSGM: Boundary formation



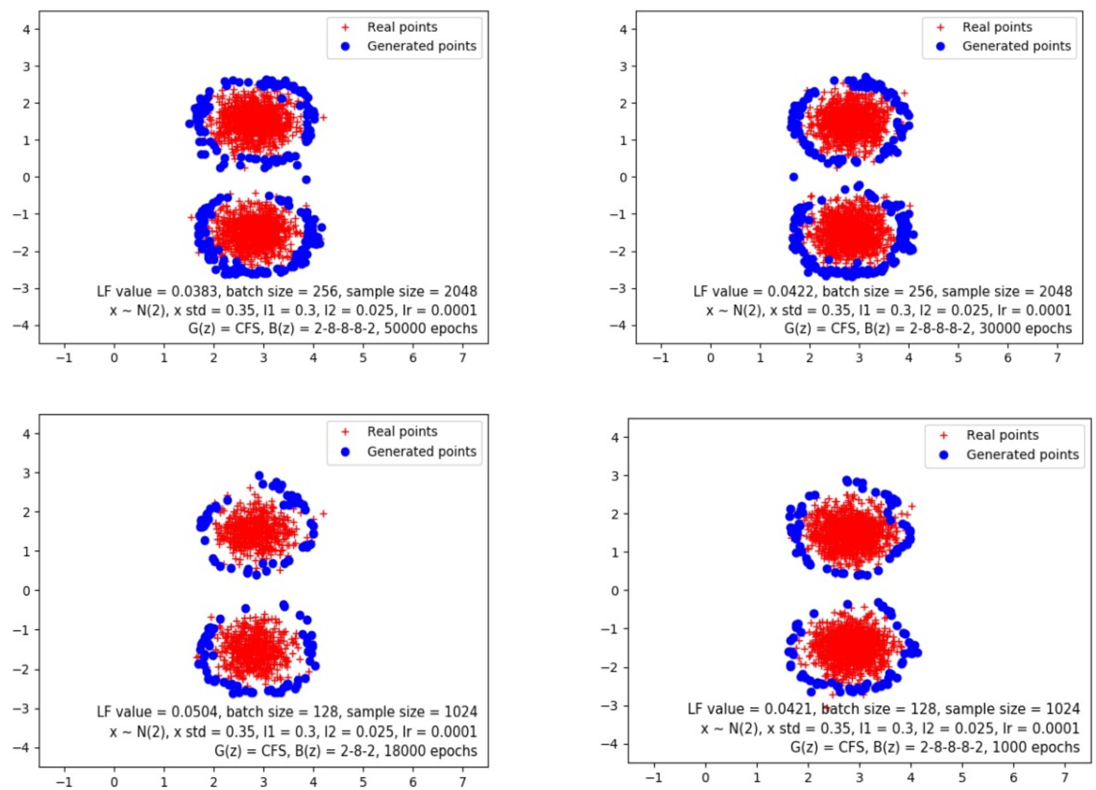
- In parallel, examine: Sample size M vs Batch size N
  - N affects boundary formation and convergence speed

Bi-Mode BDSGM Boundary Formation

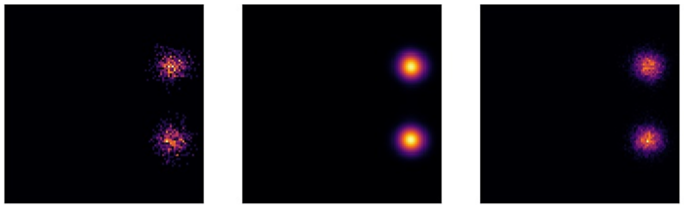
(a) IResNet- BDSGM;
B(z) FC 2-8-8-8-2



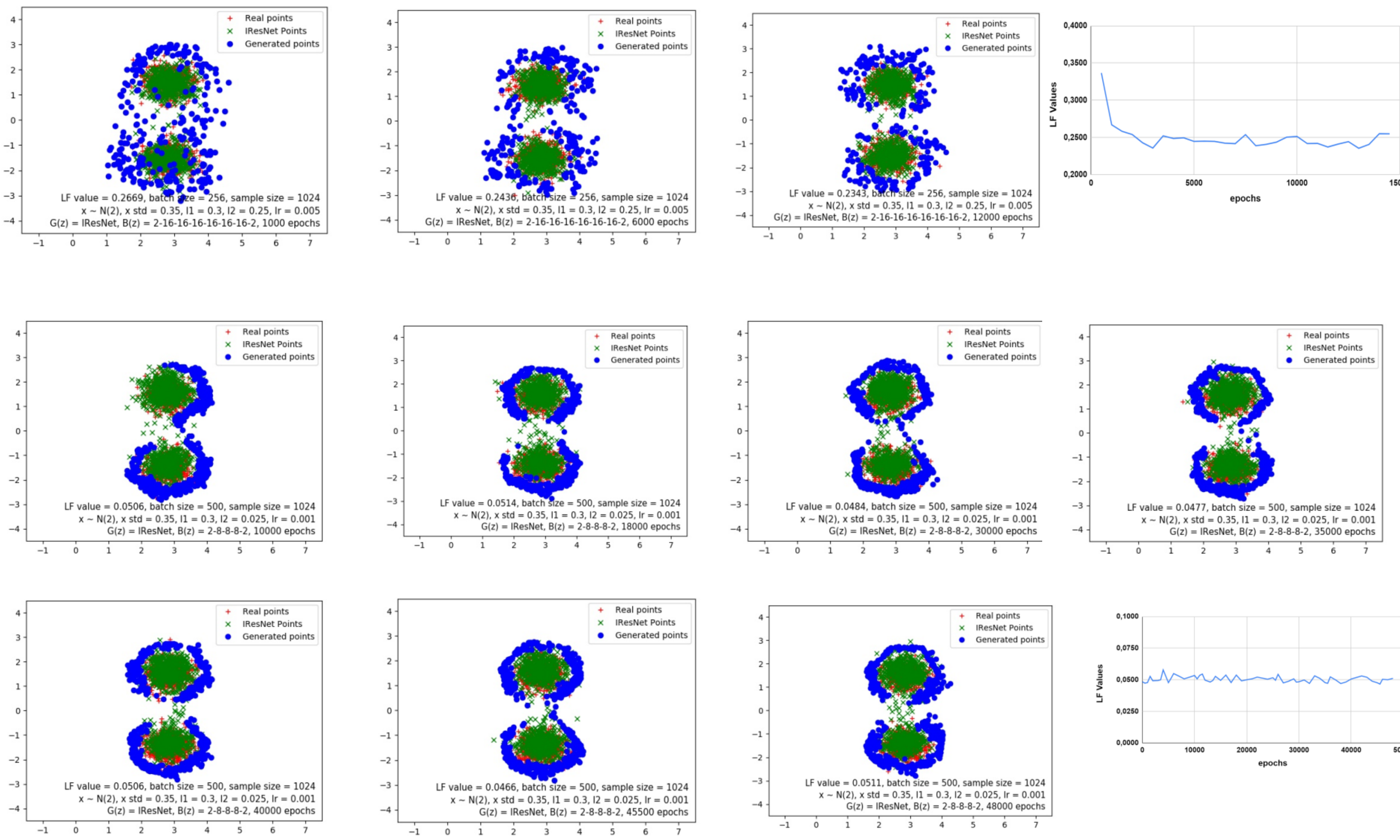
(b) CFS- BDSGM



(c) IResNet: Input samples (left), output probability density (middle), and output samples (right)



Bi-Mode Training Evolution



Conclusion

Boundary of Distribution's Support Generative Model (BDSGM):

- Train an invertible generative model, IResNet
- Create algorithm for sample generation on the boundary
- Obviate the sampling complexity problem
- Proposed loss function:
    - Forces samples to lie on the boundary
- Multi-mode distribution
    - Support: Disconnected components