Modelagem de Lentes Gravitacionais Com Normalizing Flows

Métodos em Grande Volume de Dados e Astroinformática
Professor: Clécio Roque de Bom
Aluno: Vitor Souza Ramos

Introdução

- Deformação do espaço-tempo por objetos massivos causa deflexão na luz de objetos mais distantes
- Permitem estudo da distribuição de massa em galáxias (ligado a matéria escura), medições de H_0, telescópios gravitacionais (Alto redshift), etc.
- Surveys futuros (Euclid, LSST) esperam detectar grande quantidade destes sistemas
- Análise automatizada e rápida é muito importante para a ciência feita com essa informação

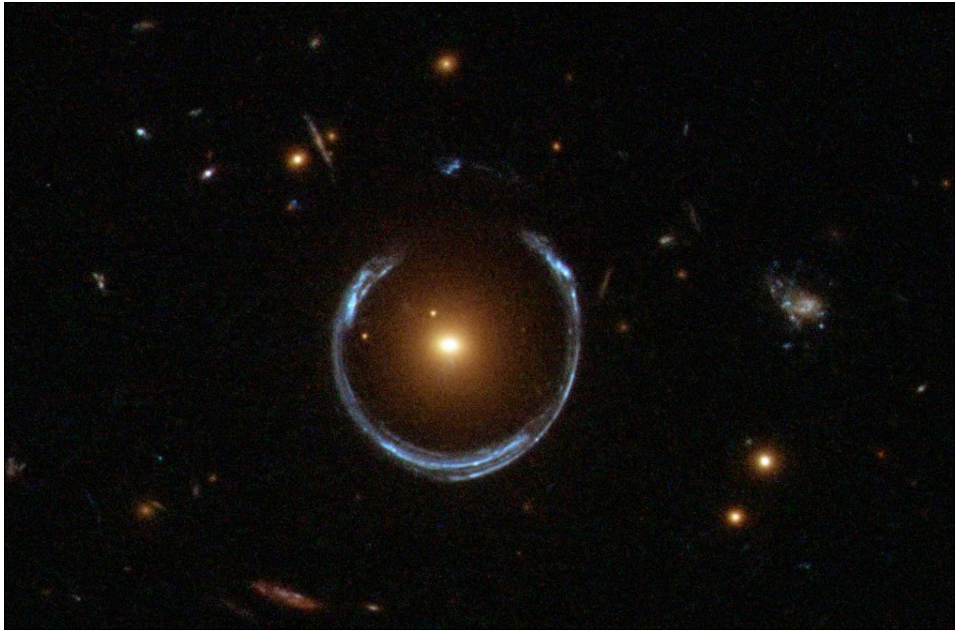


Imagem: https://en.wikipedia.org/wiki/Strong_gravitational_lensing

Metodologia - Simulation-Based Inference

- Teorema de Bayes: $$P(\theta|X) = \frac{P(\theta)P(X|\theta)}{P(X)}$$

- Likelihood é intratável → Simulation-Based Inference (SBI)

- SBI: Substituímos a likelihood por um simulador que toma parâmetros θ do prior e gera as simulações (imagens)

- Sequential Neural Posterior Estimation (SNPE): Treinamos um Estimador de Densidade para aprender a relação entre a imagem simulada e seus parâmetros

- Inferência: Mostramos a imagem ao estimador de densidade e retiramos amostras dos parâmetros calculados para reconstruir uma distribuição posterior

Metodologia - Normalizing Flows

- Normalizing Flows são uma classe de Estimadores de Densidade
- Transformações inversíveis que são usadas para mapear uma distribuição de probabilidade simples (ex. gaussiana) para uma distribuição mais complexa
- Obtemos a distribuição transformada usando a Fórmula de Mudança de variáveis
- Boas escolhas de Flows: Estratégias para transformações expressivas e tratáveis (determinante fácil de calcular)

$$z \sim p(z)$$

$$x = F(z) = f_n \circ f_{n-1} \circ \ldots \circ f_1$$

$$p(x) = p(F^{-1}(x)) \left| \det \frac{\partial F^{-1}(x)}{\partial x} \right|$$

$$p(x) = p(z) \left| \det J_{z,x} \right|$$

Metodologia - Normalizing Flows

- Masked Autoregressive Flows (MAF):

$$Z^D = \{z_1, z_2, ..., z_D\}$$
$$x_i = h(z_i : \Theta)$$
$$\Theta \sim NN(z_{1:i-1})$$

  - Jacobiano é matriz triangular inferior, determinante é produto da diagonal principal

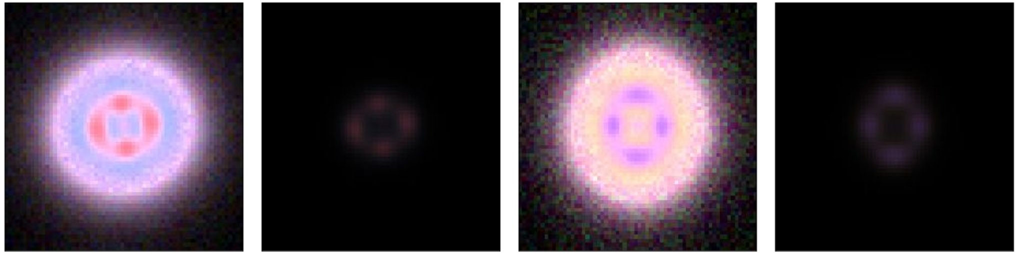
- Neural Spline Flows (NSF):
  - Mapeamento de input → Sigmoid
  - Espaço dividido em K intervalos (splines)
  - Transformação por spline
  - Alta expressividade mantendo autoregressividade

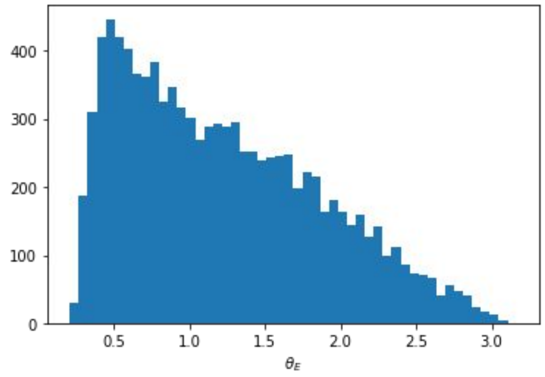
$$z_i \to [0, 1]$$
$$x_i = h^k(z_i : \Theta^K)$$

Implementação – Gerando Imagens

- Dataset gerado com pacote deeplenstronomy → 10 mil imagens
  - Resolução 64x64
  - 3 bandas (g, r, i)
  - Configurações do Survey Delve
- Priors escolhidos para gerar imagens
- Treinamento
  - Feito com 90% das imagens, 10% para teste
  - Imagens normalizadas

$$\sigma_v \sim \mathcal{U}(150, 400)$$
$$e_1 \sim \mathcal{U}(-0.4, 0.4)$$
$$e_2 \sim \mathcal{U}(-0.4, 0.4)$$
$$z_l \sim \mathcal{U}(0.2, 0.4)$$
$$z_s \sim \mathcal{U}(0.6, 0.8)$$





Implementação - Workflow

- Pacote sbi: Implementação de SNPE, MAF e NSF
- Definimos uma Embedding Net → 16 features
- Escolhemos o tipo de Estimador de Densidade → MAF ou NSF
    - Quantidade de transformações → 4
    - Hidden Units → 128
- Definimos Método de Inferência → SNPE
- Treinamento:
    - Learning Rate: 5e-4
    - Batch size: 50
    - Otimizador: Adam
    - Paciência: 20 épocas
    - Fração de validação: 10%
- Resultado: Gerador de amostras do Posterior → 5000 amostras

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 8, 64, 64]             224
              ReLU-2            [-1, 8, 64, 64]               0
         MaxPool2d-3            [-1, 8, 32, 32]               0
            Conv2d-4           [-1, 16, 32, 32]           1,168
              ReLU-5           [-1, 16, 32, 32]               0
         MaxPool2d-6           [-1, 16, 16, 16]               0
            Conv2d-7           [-1, 32, 16, 16]           4,640
              ReLU-8           [-1, 32, 16, 16]               0
         MaxPool2d-9             [-1, 32, 8, 8]               0
           Linear-10                   [-1, 32]          65,568
             ReLU-11                   [-1, 32]               0
           Linear-12                   [-1, 16]             528
             ReLU-13                   [-1, 16]               0
      FCEmbedding-14                   [-1, 16]               0
================================================================
Total params: 72,128
Trainable params: 72,128
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.05
Forward/backward pass size (MB): 0.99
Params size (MB): 0.28
Estimated Total Size (MB): 1.31
----------------------------------------------------------------
```

Implementação - Métricas de Avaliação

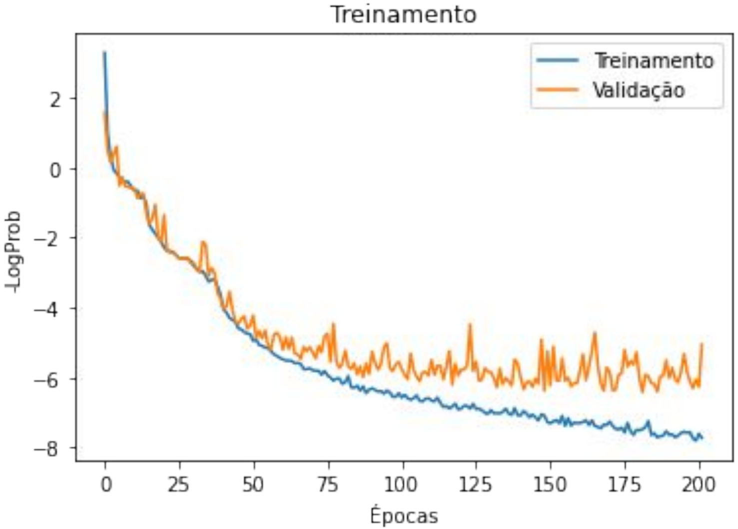
- Para uma imagem isolada:
  - Corner Plot

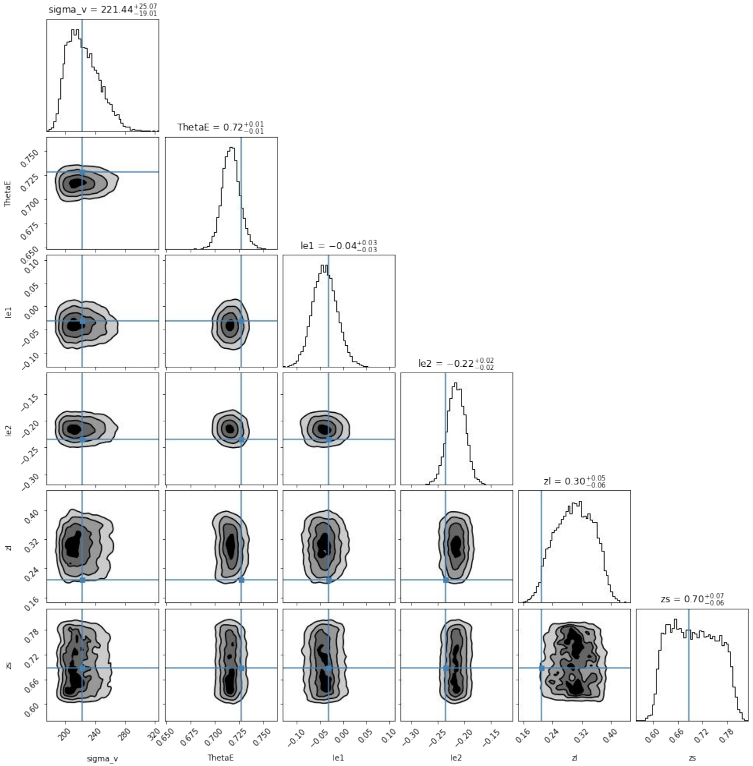
- Para o dataset de teste como um todo:
  - Plot 1x1
  - Resíduo
  - R^2

- Consideração sobre resultados
  - Imagens reais indisponíveis
  - Diversas funcionalidades não exploradas do simulador
  - Exposição do método e avaliação da capacidade de modelar uma distribuição de probabilidade

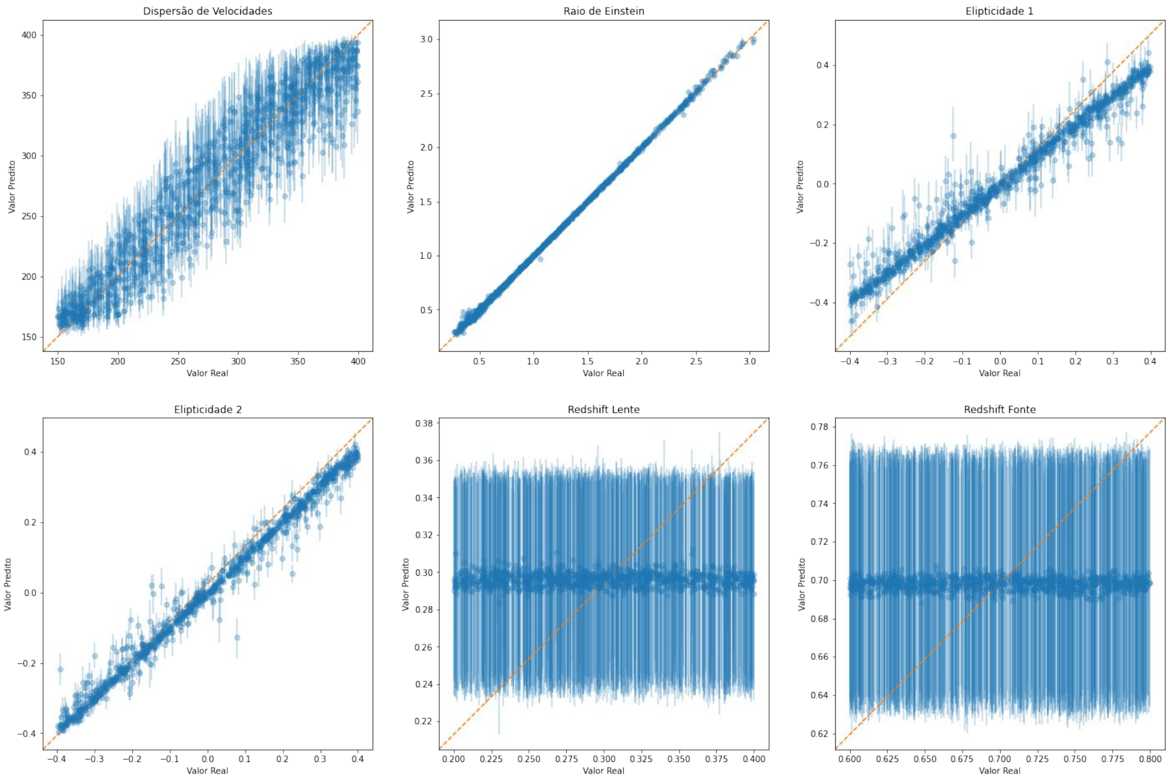
Resultados - MAF

- Treinamento
  - 202 épocas
  - Aprox. 8 minutos



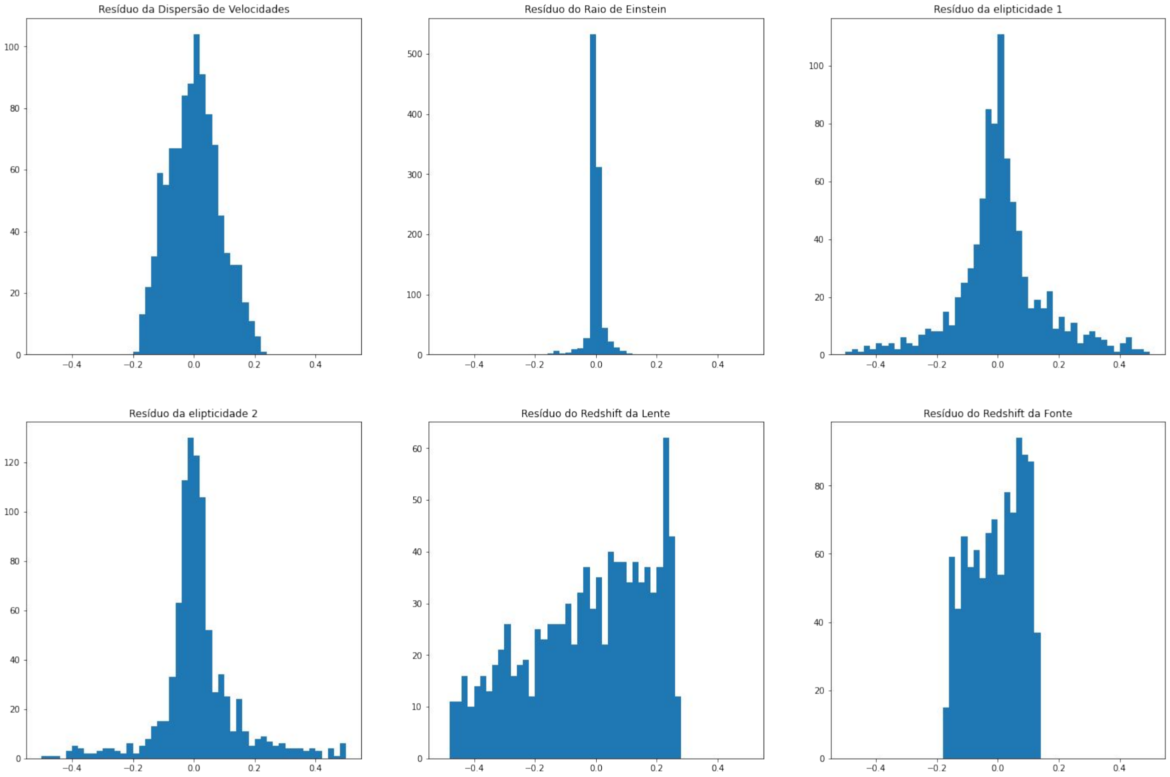


Resultados – MAF



R^2	
σ_v	0.903
θ_E	0.999
e_1	0.976
e_2	0.987
z_l	-0.007
z_s	-0.006

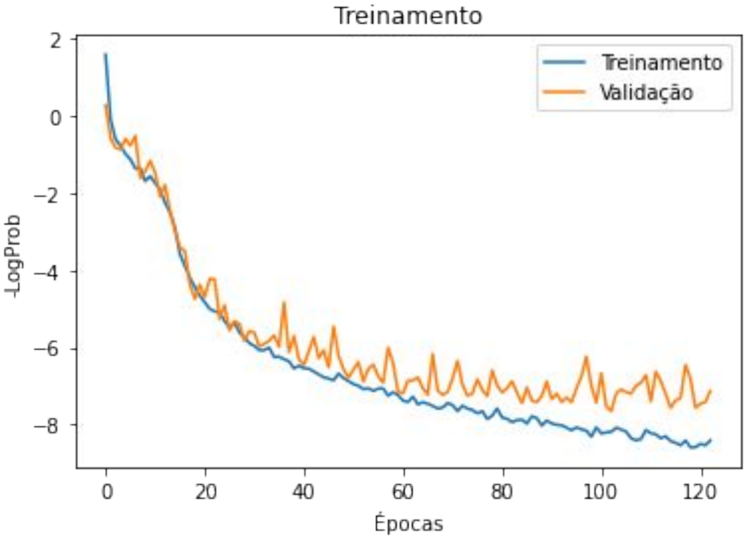
Resultados – MAF

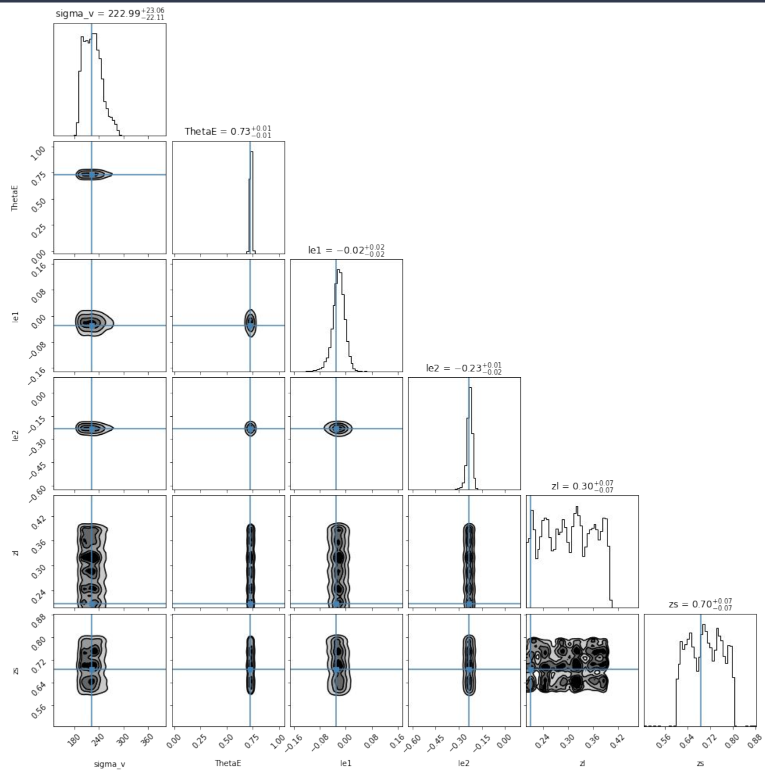


Resíduo Abs. Médio	
σ_v	0.067
θ_E	0.014
e_1	0.269
e_2	0.179
z_l	0.171
z_s	0.0742

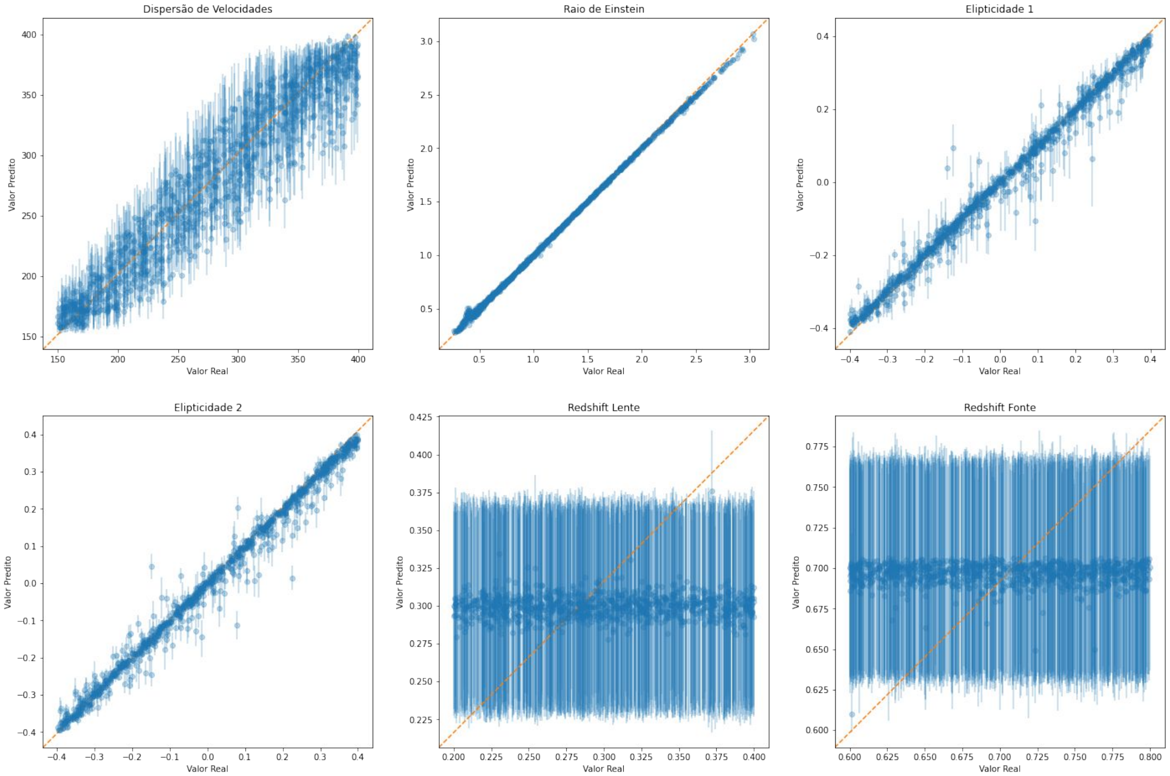
Resultados - NSF

- Treinamento
  - 123 épocas
  - Aprox. 15 minutos



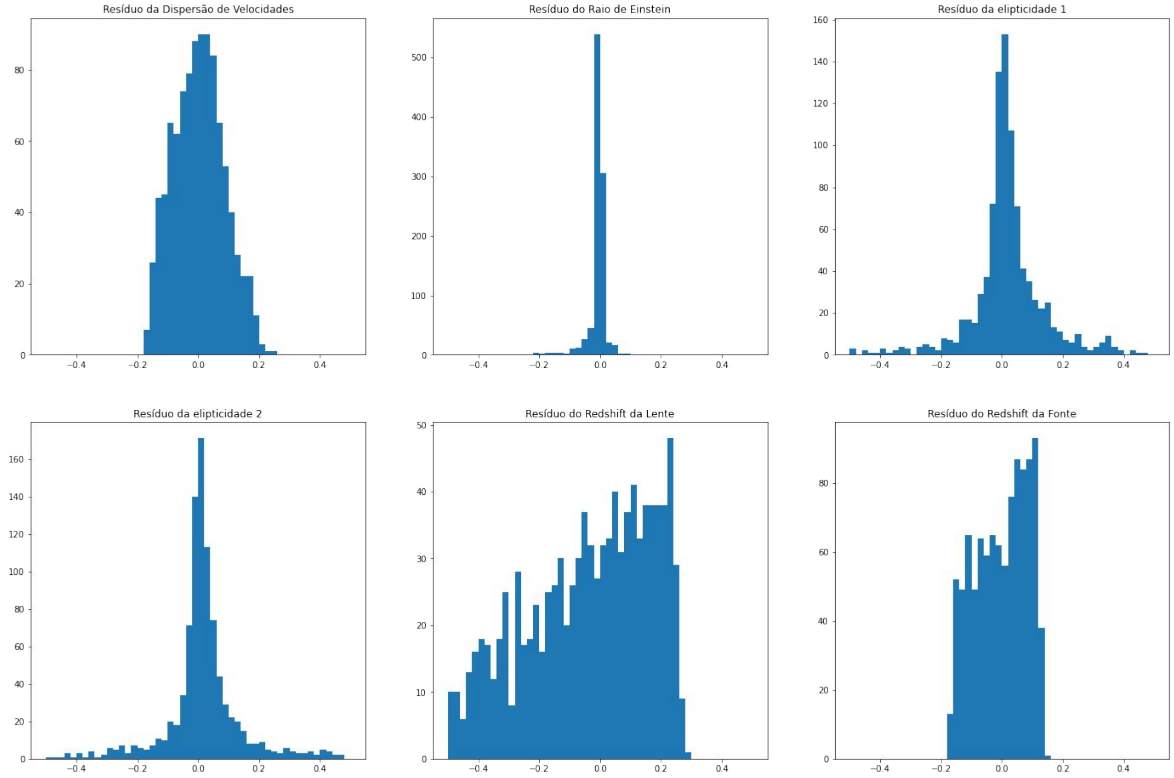


Resultados – NSF



R^2	
σ_v	0.902
θ_E	0.999
e_1	0.988
e_2	0.988
z_l	-0.008
z_s	-0.012

Resultados – NSF



Resíduo Abs. Médio	
σ_v	0.068
θ_E	0.014
e_1	0.204
e_2	0.173
z_l	0.173
z_s	0.074

Conclusão

- Os dois estimadores de densidade foram capazes de aprender a modelar bem o raio de Einstein
- MAF teve viés nas elipticidades, NSF foi mais expressivo
- Apesar da alta variação a dispersão de velocidades também é razoavelmente bem modelada
- NSF demora mais para treinar, mas evitou erro viés nas elipticidades
- Nenhum dos dois Estimadores de Densidade foi capaz de modelar os redshifts → Possível falha nas simulações
- NSF revisitado → 16 Transf., 512 H.U.: Resultados similares
- Possibilidades Futuras
    - Investigar simulações
    - Exploração de outros métodos de inferência (SNLE, SNRE)
    - Outros tipos de transformações (MDN)
    - Aplicação de modelos a imagens reais

Referências

1. [1911.01429] The frontier of simulation-based inference
2. [1908.09257] Normalizing Flows: An Introduction and Review of Current Methods
3. [1505.05770] Variational Inference with Normalizing Flows
4. [1905.07488] Automatic Posterior Transformation for Likelihood-Free Inference
5. [1705.07057] Masked Autoregressive Flow for Density Estimation
6. [1906.04032] Neural Spline Flows
7. [1502.03509] MADE: Masked Autoencoder for Distribution Estimation
8. [2102.02830] deeplenstronomy: A dataset simulation package for strong gravitational lensing
9. [1911.06341] Deep Learning in Wide-field Surveys: Fast Analysis of Strong Lenses in Ground-based Cosmic Experiments
10. [2210.10793] LeMoN: Lens Modelling with Neural networks -- I. Automated modelling of strong gravitational lenses with Bayesian Neural Networks

Referências

11 [2205.09126] Estimating the warm dark matter mass from strong lensing images with truncated marginal neural ratio estimation

12 Simulation-Based Inference of Strong Gravitational Lensing Parameters - https://arxiv.org/abs/2112.05278

13 Strong Lensing Parameter Estimation on Ground-Based Imaging Data Using Simulation-Based Inference - https://arxiv.org/abs/2211.05836

14 Normalizing Flows for Probabilistic Modeling and Inference - https://arxiv.org/abs/1912.02762

15 Pacote sbi - https://www.mackelab.org/sbi

16 https://akosiorek.github.io/ml/2018/04/03/norm_flows.html