Flow-based likelihoods for non-Gaussian inference

Ana Díaz Rivero
Harvard University

arXiv: 2007.05535 (submitted to PRD, Diaz Rivero & Dvorkin 2020)

Outline

1. Gaussian likelihoods
2. Data-driven likelihoods:
    a. Gaussian mixture models
    b. Independent component analysis
    c. **Flow-based likelihoods**
3. Quantifying non-Gaussianity in a dataset
4. Application to the weak lensing convergence power spectrum
5. (Application to the galaxy power spectrum)
6. Summary and conclusions

Likelihoods

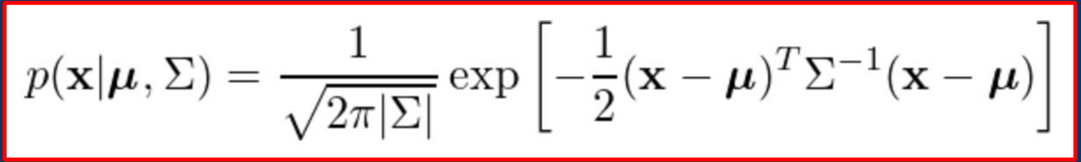
The likelihood measures the extent to which a sample provides support for particular values in a statistical model.

Much of statistical inference is predicated on the likelihood:

- Maximum likelihood estimates
- likelihood ratio
- posteriors
- Bayes factor
- …

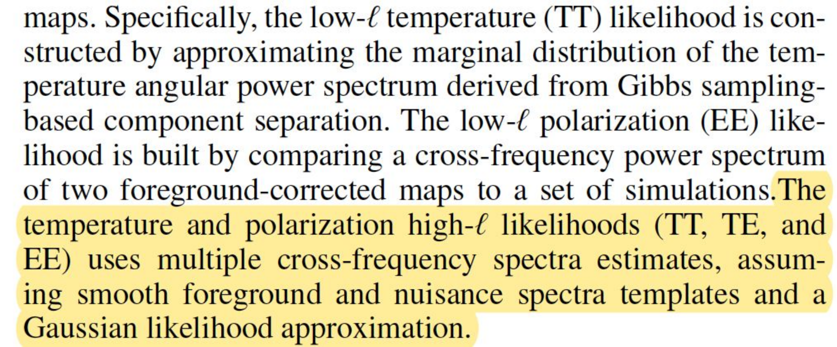
Gaussian likelihoods

Gaussian likelihoods are very widespread: well understood, only need a covariance matrix, CLT…



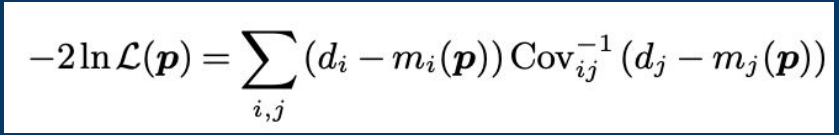
$$p(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right]$$

CMB power spectra (*Planck* 2018)



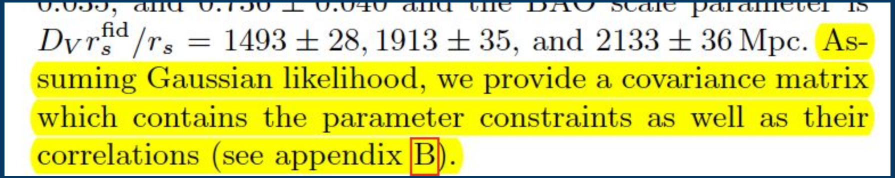
maps. Specifically, the low-ℓ temperature (TT) likelihood is constructed by approximating the marginal distribution of the temperature angular power spectrum derived from Gibbs sampling-based component separation. The low-ℓ polarization (EE) likelihood is built by comparing a cross-frequency power spectrum of two foreground-corrected maps to a set of simulations.The temperature and polarization high-ℓ likelihoods (TT, TE, and EE) uses multiple cross-frequency spectra estimates, assuming smooth foreground and nuisance spectra templates and a Gaussian likelihood approximation.

Shear 2pt function (HSC)



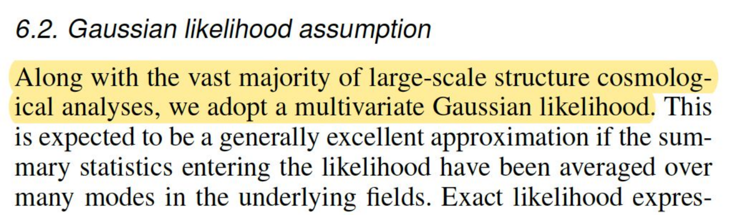
$$-2\ln\mathcal{L}(\boldsymbol{p}) = \sum_{i,j} (d_i - m_i(\boldsymbol{p}))\,\mathrm{Cov}^{-1}_{ij}\,(d_j - m_j(\boldsymbol{p}))$$

Galaxy power spectrum (SDSS-III BOSS)



0.035, and 0.736 ± 0.040 and the BAO scale parameter is $D_V r_s^{\mathrm{fid}}/r_s = 1493 \pm 28, 1913 \pm 35,$ and 2133 ± 36 Mpc. Assuming Gaussian likelihood, we provide a covariance matrix which contains the parameter constraints as well as their correlations (see appendix B).

Galaxy clustering + weak lensing (KiDS-1000)



6.2. Gaussian likelihood assumption

Along with the vast majority of large-scale structure cosmological analyses, we adopt a multivariate Gaussian likelihood. This is expected to be a generally excellent approximation if the summary statistics entering the likelihood have been averaged over many modes in the underlying fields. Exact likelihood expres-

Gaussian likelihoods

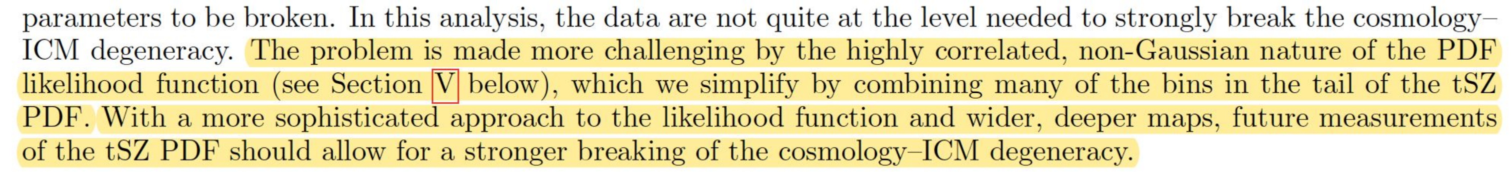
However,

- CLT isn’t always applicable (e.g. power spectra at small wavenumbers)
- If the covariance matrix is an estimated quantity, have to marginalize over the true covariance (Gaussian → *t*-distribution)
- Systematic effects can introduce non-Gaussian correlations
- Physics giving rise to an observable: a nonlinear function of Gaussian RVs is not Gaussian distributed (CMB vs. galaxy distributions, cosmic shear)

Gaussian likelihoods

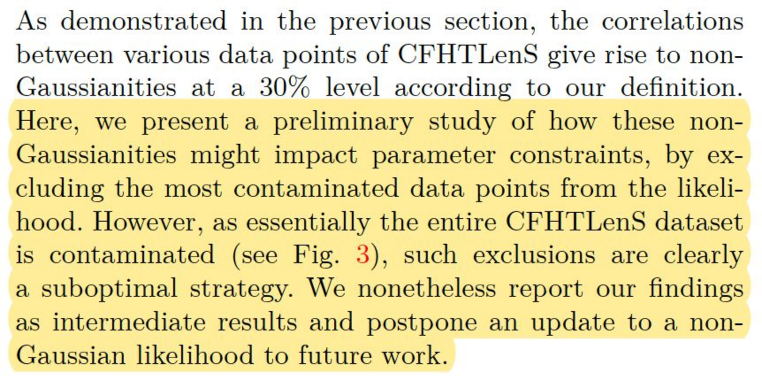
There isn’t always a clear alternative/better likelihood:

ACT Thermal SZ one-point PDF (Hill+, 2015)



parameters to be broken. In this analysis, the data are not quite at the level needed to strongly break the cosmology–ICM degeneracy. The problem is made more challenging by the highly correlated, non-Gaussian nature of the PDF likelihood function (see Section V below), which we simplify by combining many of the bins in the tail of the tSZ PDF. With a more sophisticated approach to the likelihood function and wider, deeper maps, future measurements of the tSZ PDF should allow for a stronger breaking of the cosmology–ICM degeneracy.

CFHTLenS shear correlation (Sellentin+, 2018)



As demonstrated in the previous section, the correlations between various data points of CFHTLenS give rise to non-Gaussianities at a 30% level according to our definition. Here, we present a preliminary study of how these non-Gaussianities might impact parameter constraints, by excluding the most contaminated data points from the likelihood. However, as essentially the entire CFHTLenS dataset is contaminated (see Fig. 3), such exclusions are clearly a suboptimal strategy. We nonetheless report our findings as intermediate results and postpone an update to a non-Gaussian likelihood to future work.

An alternative: data-driven likelihoods (DDLs)

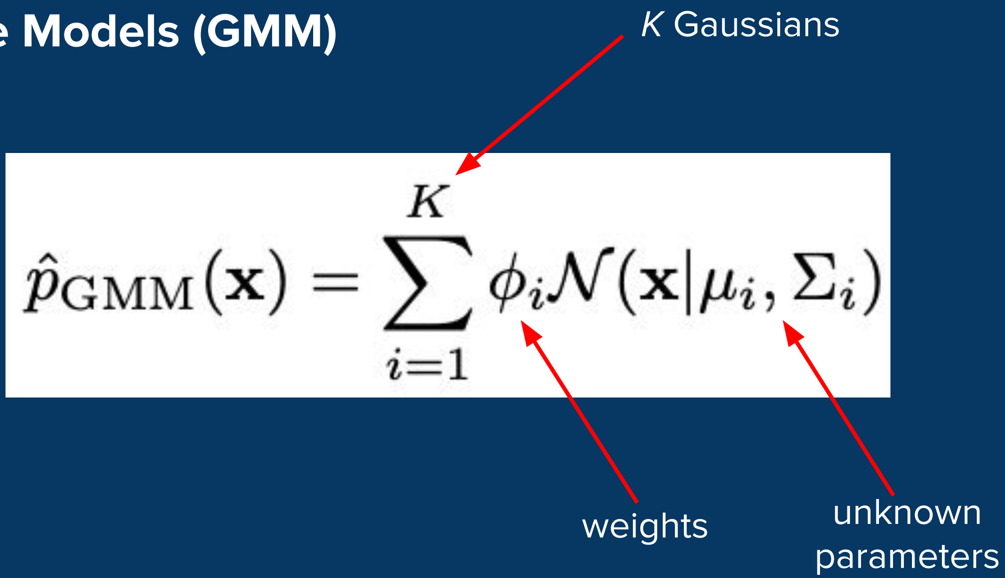
Data-driven likelihoods are learned directly from the data:

- We can think of mock data* as independent draws from the underlying true likelihood function.
- We can estimate the data's PDF with sufficient samples from it.

The hope is that DDLs can accurately capture non-Gaussianities in the data.

An alternative: data-driven likelihoods (DDLs)

Gaussian Mixture Models (GMM)



$$\hat{p}_{\mathrm{GMM}}(\mathbf{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i)$$

K Gaussians

weights

unknown parameters

Use expectation maximization to find parameters, BIC to determine *K*.

An alternative: data-driven likelihoods (DDLs)

Independent Component Analysis (ICA)

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

1. Decorrelate data (project onto PCs)
2. Normalize PCs
3. Rotate components to maximize independence

$$\hat{\mathbf{s}} \equiv \mathbf{x}_{\mathrm{ICA}} = \mathbf{W}\mathbf{x} = \{\mathbf{x}_{1,\mathrm{ICA}}, \ldots, \mathbf{x}_{N,\mathrm{ICA}}\}$$

$$\Big\downarrow \ \text{KDE}$$

$$\hat{p}_{\mathrm{ICA}}(\mathbf{x}) = \prod_{n=1}^{N} \hat{p}_n(\mathbf{x})$$

An alternative: data-driven likelihoods (DDLs)

Flow-based Likelihoods (FBLs, Diaz Rivero & Dvorkin 2020)

→ I will introduce flow-based generative models (esp. FFJORD)

→ Their minimization objective is what we will call a flow-based likelihood

Flow-based generative models

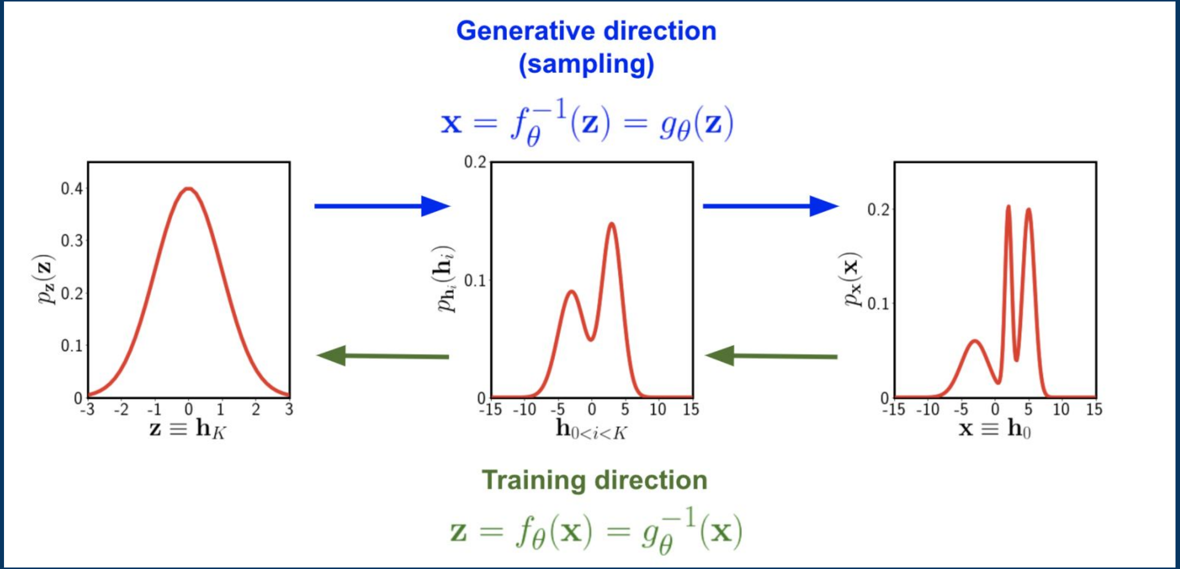
Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).

Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

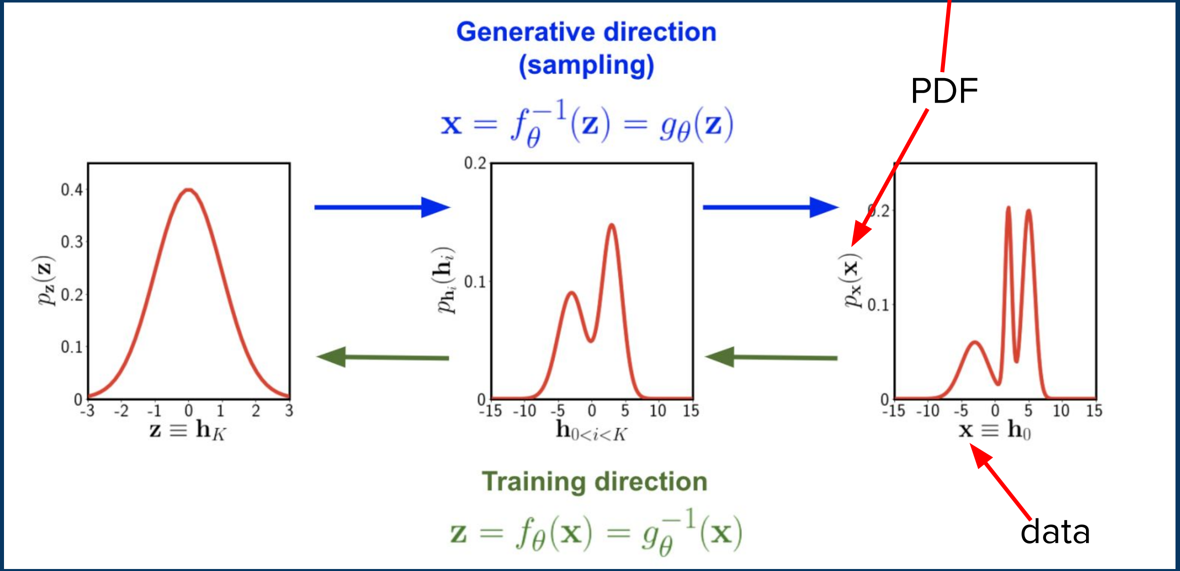
In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).



Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).

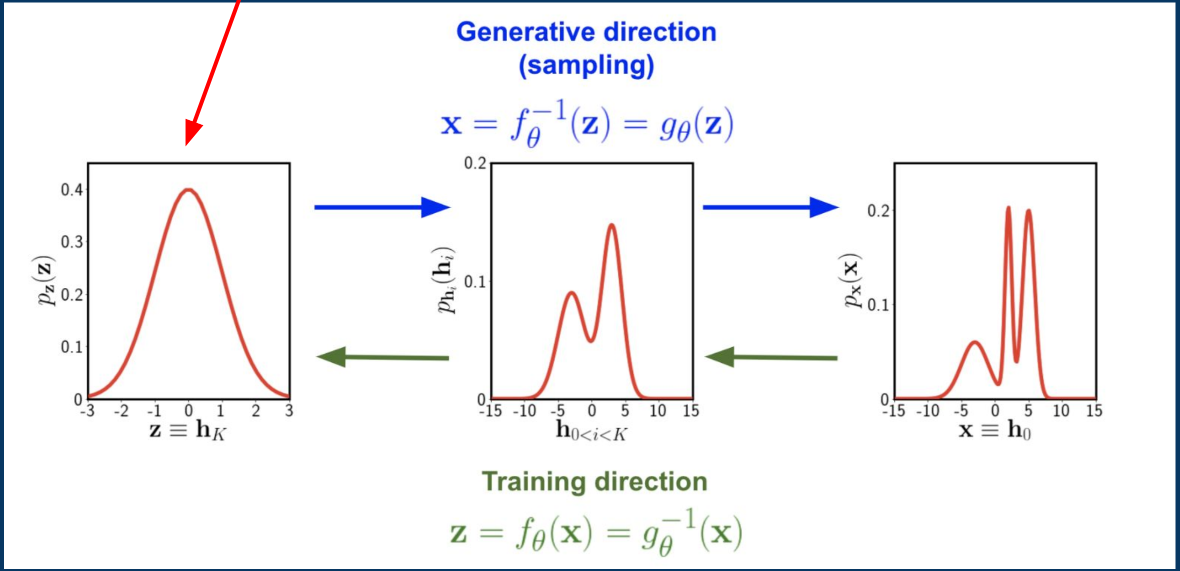


Generative direction (sampling)
$\mathbf{x} = f_\theta^{-1}(\mathbf{z}) = g_\theta(\mathbf{z})$
$p_\mathbf{z}(\mathbf{z})$
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3
$\mathbf{z} \equiv \mathbf{h}_K$
$p_{\mathbf{h}_i}(\mathbf{h}_i)$
0.2
0.1
0
-15
-10
-5
0
5
10
15
$\mathbf{h}_{0<i<K}$
PDF
$p_\mathbf{x}(\mathbf{x})$
0.2
0.1
0
-15
-10
-5
0
5
10
15
$\mathbf{x} \equiv \mathbf{h}_0$
data
Training direction
$\mathbf{z} = f_\theta(\mathbf{x}) = g_\theta^{-1}(\mathbf{x})$

Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).



Generative direction (sampling)

$$\mathbf{x} = f_\theta^{-1}(\mathbf{z}) = g_\theta(\mathbf{z})$$

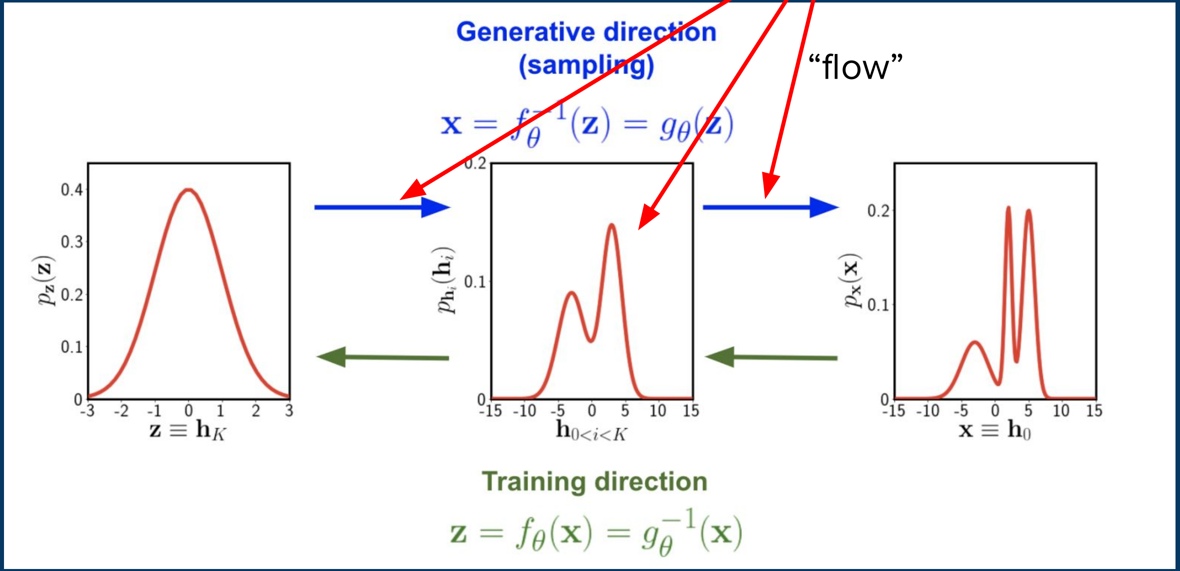
Training direction

$$\mathbf{z} = f_\theta(\mathbf{x}) = g_\theta^{-1}(\mathbf{x})$$

Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).

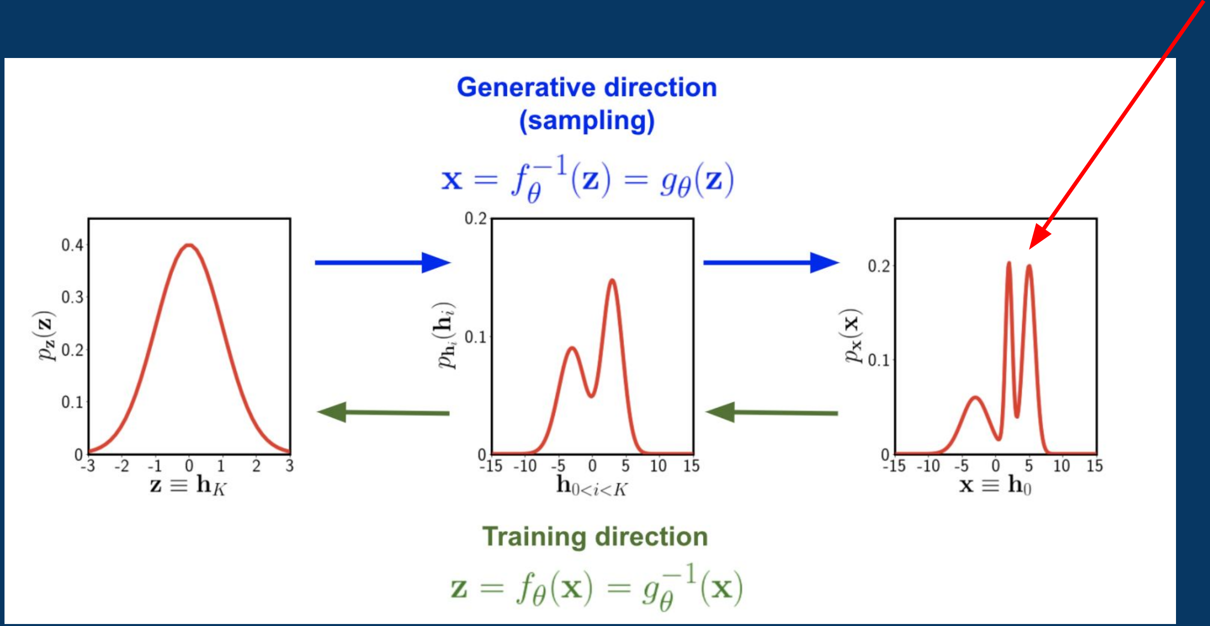

Generative direction
(sampling)
$\mathbf{x} = f_\theta^{-1}(\mathbf{z}) = g_\theta(\mathbf{z})$
$p_{\mathbf{z}}(\mathbf{z})$
0.4
0.3
0.2
0.1
0
-3 -2 -1 0 1 2 3
$\mathbf{z} \equiv \mathbf{h}_K$
$p_{\mathbf{h}_i}(\mathbf{h}_i)$
0.2
0.1
0
-15 -10 -5 0 5 10 15
$\mathbf{h}_{0<i<K}$
$p_{\mathbf{x}}(\mathbf{x})$
0.2
0.1
0
-15 -10 -5 0 5 10 15
$\mathbf{x} \equiv \mathbf{h}_0$
Training direction
$\mathbf{z} = f_\theta(\mathbf{x}) = g_\theta^{-1}(\mathbf{x})$

“flow”

Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).



Generative direction
(sampling)
$\mathbf{x} = f_\theta^{-1}(\mathbf{z}) = g_\theta(\mathbf{z})$
$p_{\mathbf{z}}(\mathbf{z})$
0.4
0.3
0.2
0.1
0
-3 -2 -1 0 1 2 3
$\mathbf{z} \equiv \mathbf{h}_K$
$p_{\mathbf{h}_i}(\mathbf{h}_i)$
0.2
0.1
0
-15 -10 -5 0 5 10 15
$\mathbf{h}_{0<i<K}$
$p_{\mathbf{x}}(\mathbf{x})$
0.2
0.1
0
-15 -10 -5 0 5 10 15
$\mathbf{x} \equiv \mathbf{h}_0$
Training direction
$\mathbf{z} = f_\theta(\mathbf{x}) = g_\theta^{-1}(\mathbf{x})$

Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).

$$\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$$

$$\mathbf{x} \equiv \mathbf{h}_0 \xleftrightarrow[g_K]{f_1} \mathbf{h_1} \xleftrightarrow[g_{K-1}]{f_2} \dots \xleftrightarrow[g_2]{f_{K-1}} \mathbf{h_{K-1}} \xleftrightarrow[g_1]{f_K} \mathbf{h_K} \equiv \mathbf{z}$$

$$f \;=\; f_1 \circ f_2 \circ \dots f_K$$

$$g \;=\; g_1 \circ g_2 \circ \dots g_K$$

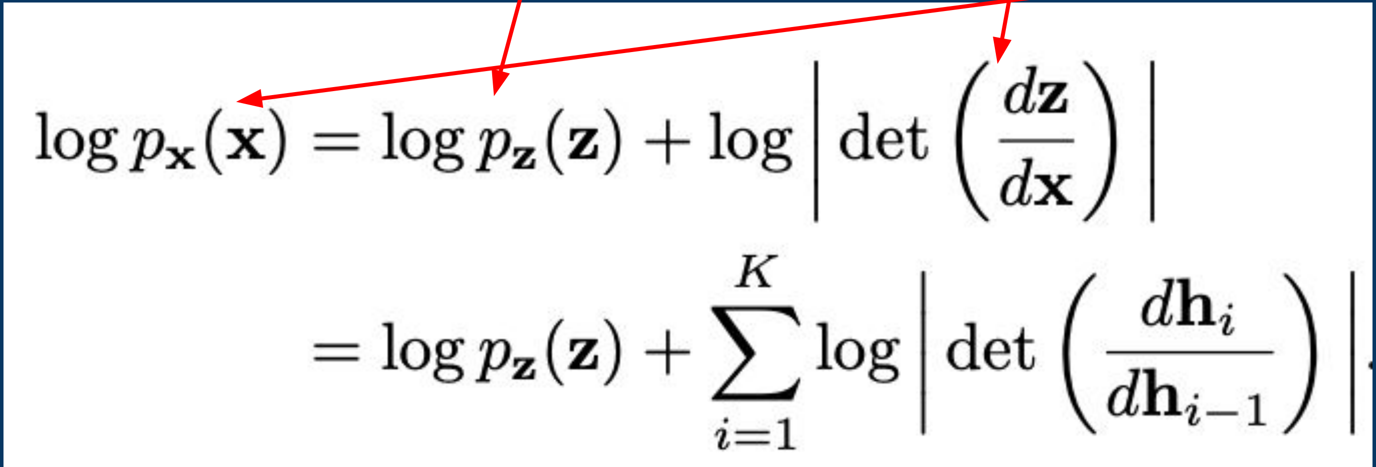
$$\mathbf{x} = g_\theta(\mathbf{z}) = f_\theta^{-1}(\mathbf{z})$$

$$\mathbf{z} = g_\theta^{-1}(\mathbf{x}) = f_\theta(\mathbf{x})$$

Flow-based generative models

Generative models aim to learn the probability distribution that gave rise to data ***x***, such that new samples can be drawn.

In *flow-based models*, a simple distribution is repeatedly transformed to match p(***x***).

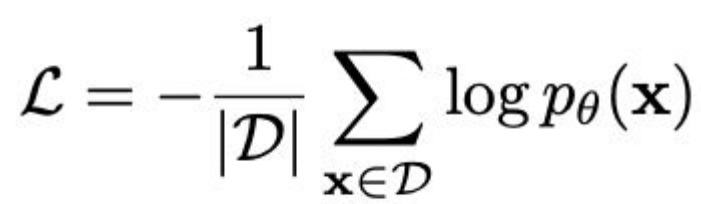


$$
\begin{aligned}
\log p_{\mathbf{x}}(\mathbf{x}) &= \log p_{\mathbf{z}}(\mathbf{z}) + \log \left| \det \left(\frac{d\mathbf{z}}{d\mathbf{x}} \right) \right| \\
&= \log p_{\mathbf{z}}(\mathbf{z}) + \sum_{i=1}^{K} \log \left| \det \left(\frac{d\mathbf{h}_i}{d\mathbf{h}_{i-1}} \right) \right|.
\end{aligned}
$$

Flow-based generative models

The goal is to train a model to learn these transformations.

- Transformations can involve (invertible) neural networks to make them very expressive
- The loss is the negative log-likelihood over the training set.



$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log p_\theta(\mathbf{x})$$

If training is successful, the learned likelihood == the true data likelihood == a DDL.

Flow-based generative models

BUT, transformations must

- be easily invertible,
- have an easy-to-compute Jacobian determinant (scales as n^3),

which limits their expressivity.

Different tricks in the literature:

- Restrict the form of the transformation to exploit identities
- Make Jacobian triangular by making transformations auto-regressive or splitting up dimensions and applying affine transformations

Ideally also want quick density estimation **and** sampling.

Flow-based generative models

Model performance is generally judged by bits per dimension (average negative log-likelihood) + qualitative sample assessments.

Glow (Kingma & Dhariwal 2018)

Time-permitting, I will show tests we ran to determine the relationship between sample quality and likelihood quality.

FFJORD (Grathwohl+ 2018)

Transformation from prior to data is seen as evolution in time.

Don't have to restrict the form of the Jacobian → very expressive.

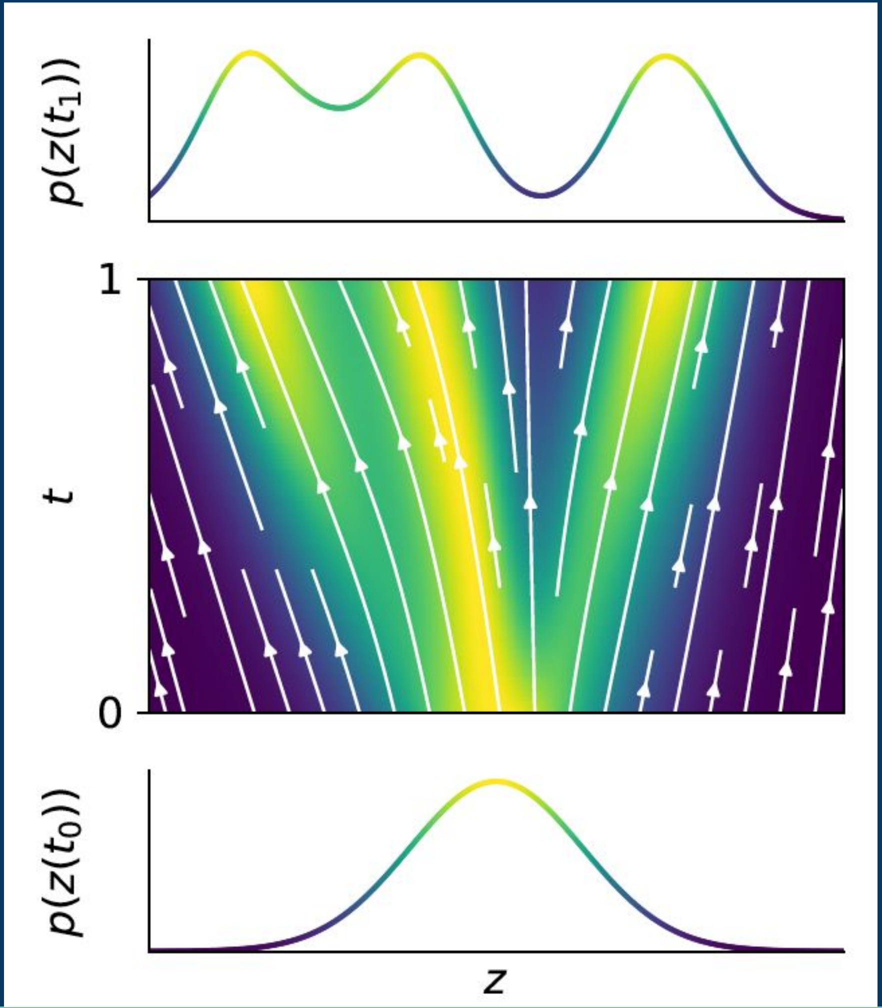
$$\frac{\partial \mathbf{z}(t)}{\partial t} = f(\mathbf{z}(t), t; \theta)$$

$$\frac{\partial \mathrm{log} p(\mathbf{z}(t))}{\partial t} = -\mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right)$$

$$\log p(\mathbf{z}(t_1)) = \mathrm{log} p(\mathbf{z}(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right) dt.$$

data

prior



FFJORD (Grathwohl+ 2018)

Transformation from prior to data is seen as evolution in time.

Don't have to restrict the form of the Jacobian → very expressive.

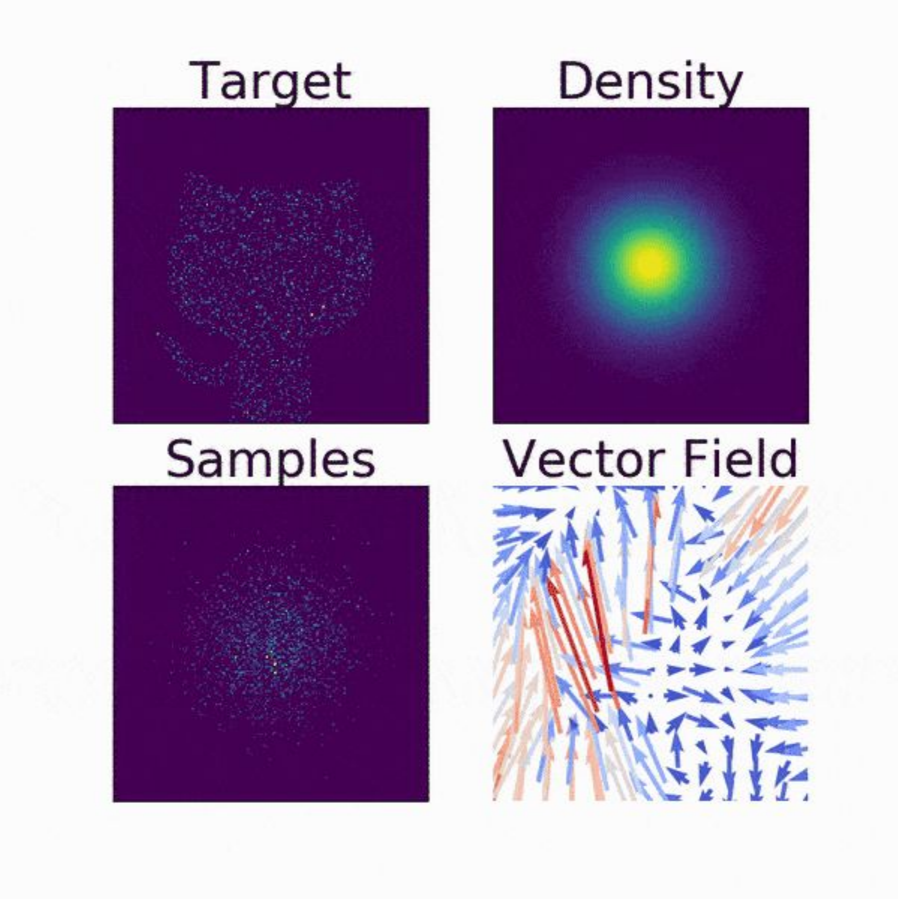
$$\frac{\partial \mathbf{z}(t)}{\partial t} = f(\mathbf{z}(t), t; \theta)$$

$$\frac{\partial \mathrm{log} p(\mathbf{z}(t))}{\partial t} = -\mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right)$$

$$\log p(\mathbf{z}(t_1)) = \mathrm{log} p(\mathbf{z}(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right) dt.$$

data

prior



Quantifying NG in a dataset

We propose identifying non-Gaussianities (NG) in three ways:

1. ***t*-statistic of skewness and excess kurtosis for every bin in the data**

$$t = \frac{\hat{\beta} - \beta_{\text{null}}}{\text{SE}(\hat{\beta})}$$

0 for a Gaussian

Quantifying NG in a dataset

We propose identifying non-Gaussianities (NG) in three ways:

2. **Transcovariance matrix (Sellentin+ 2018)**, which considers the Gaussianity of all pairs of data points

$$s_i^{u,v} = x_i^u + x_i^v$$

Should be equal for whitened Gaussian data

$$\frac{1}{b}\sum_{a=1}^{b}[\mathcal{K}(s_i^{u,v}) - \mathcal{N}(0,2)]^2 \equiv S^+_{u,v}$$

Total non-Gaussian contamination for each bin

$$\epsilon_u^+ = \sum_{v\neq u} S^+_{u,v}$$

Quantifying NG in a dataset

We propose identifying non-Gaussianities (NG) in three ways:

3. **KL divergence of (the data w.r.t. a MVN)** vs (MVN with itself) (Wang+ 2009)

$$D_{n,m}(p||q) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x}$$

$$\hat{D}_{n,m}(p||q) = \frac{d}{n} \sum_{i=1}^{n} \log \frac{\nu_k(i)}{\rho_k(i)} + \log \frac{m}{n-1}$$

Unbiased kNN estimator

Quantifying NG in a dataset

Our method is going to consist of:

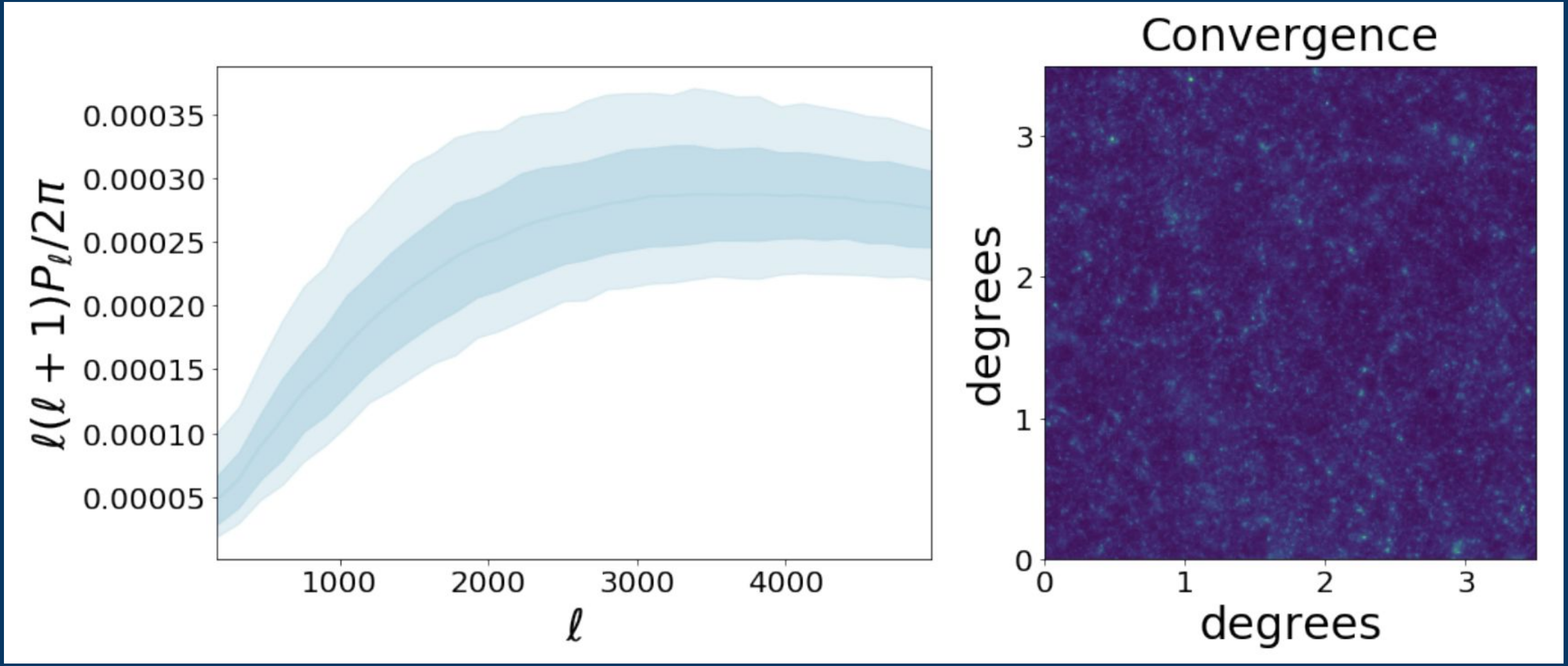
1. Applying these three tests to a mock dataset to look at the different ways in which NG can manifest themselves.

2. Generating samples from the three DDLs to assess whether each likelihood has successfully captured the NGs.

Simulated data

We consider the weak lensing convergence power spectrum.*

Simulated 75,000 mock convergence maps using LensTools (Petri 2016):



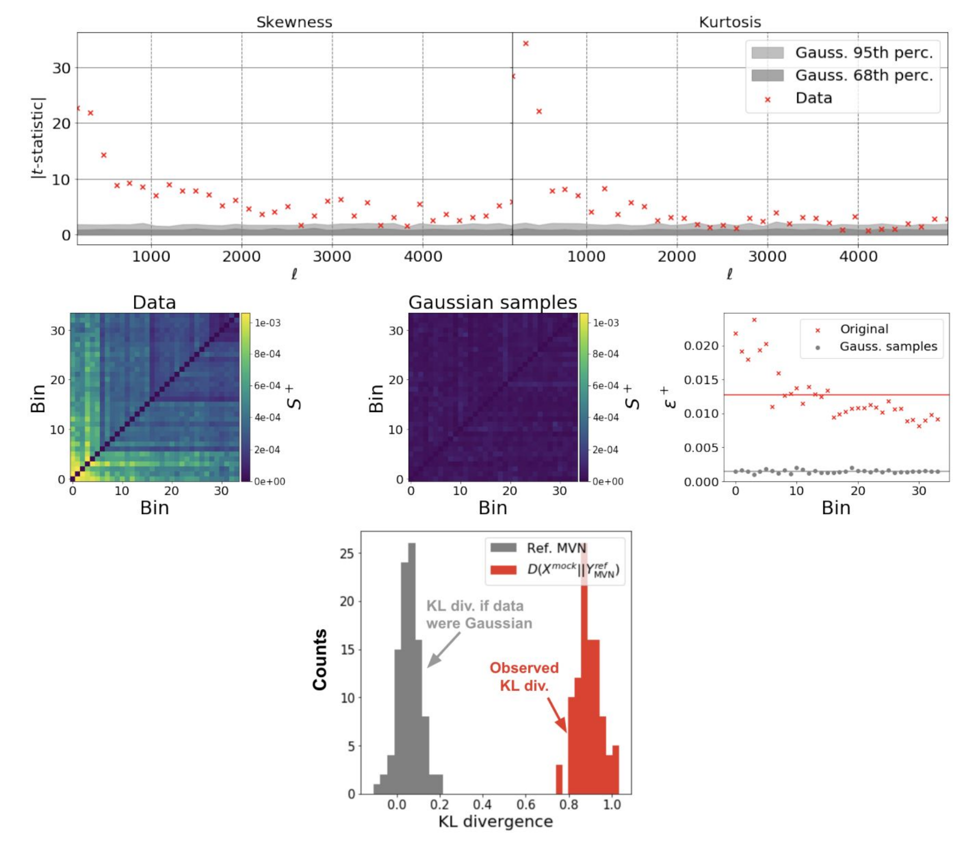
* time-permitting will show galaxy power spectrum too

NG in the weak lensing convergence power spectrum

Test 1: t-stat of skewness and kurtosis

Test 2: transcovariance matrix

Test 3: KL divergence

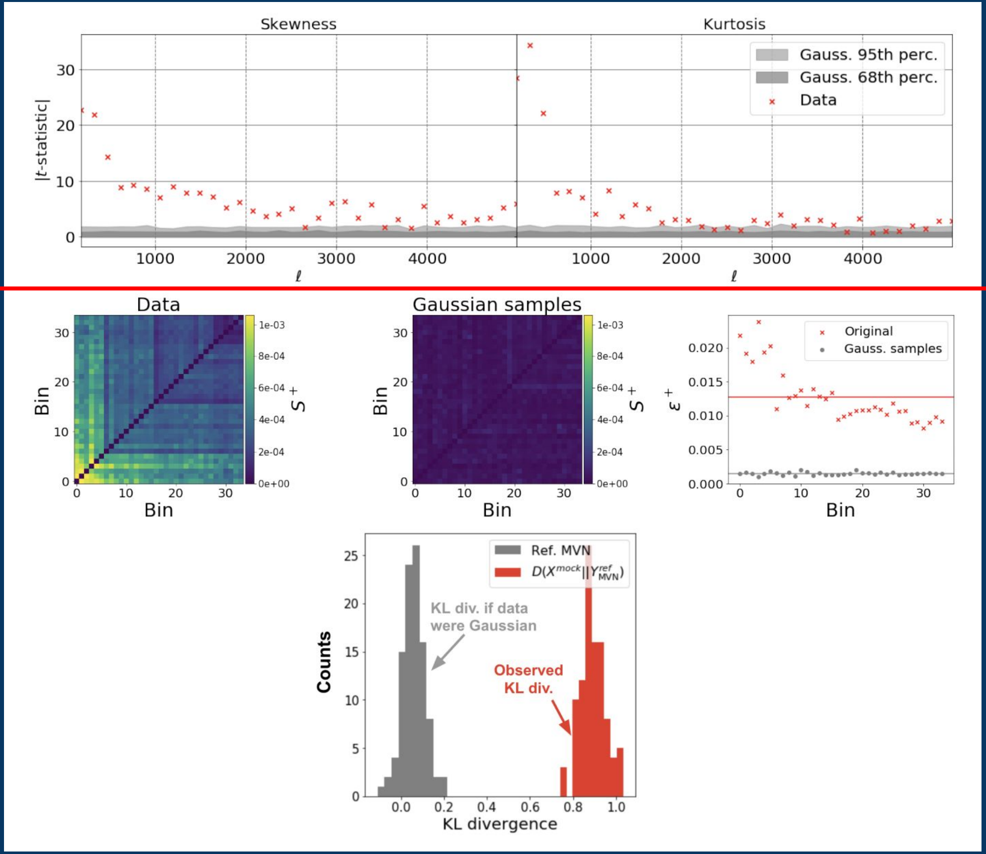


NG in the weak lensing convergence power spectrum

Test 1: t-stat of skewness and kurtosis

Test 2: transcovariance matrix

Test 3: KL divergence

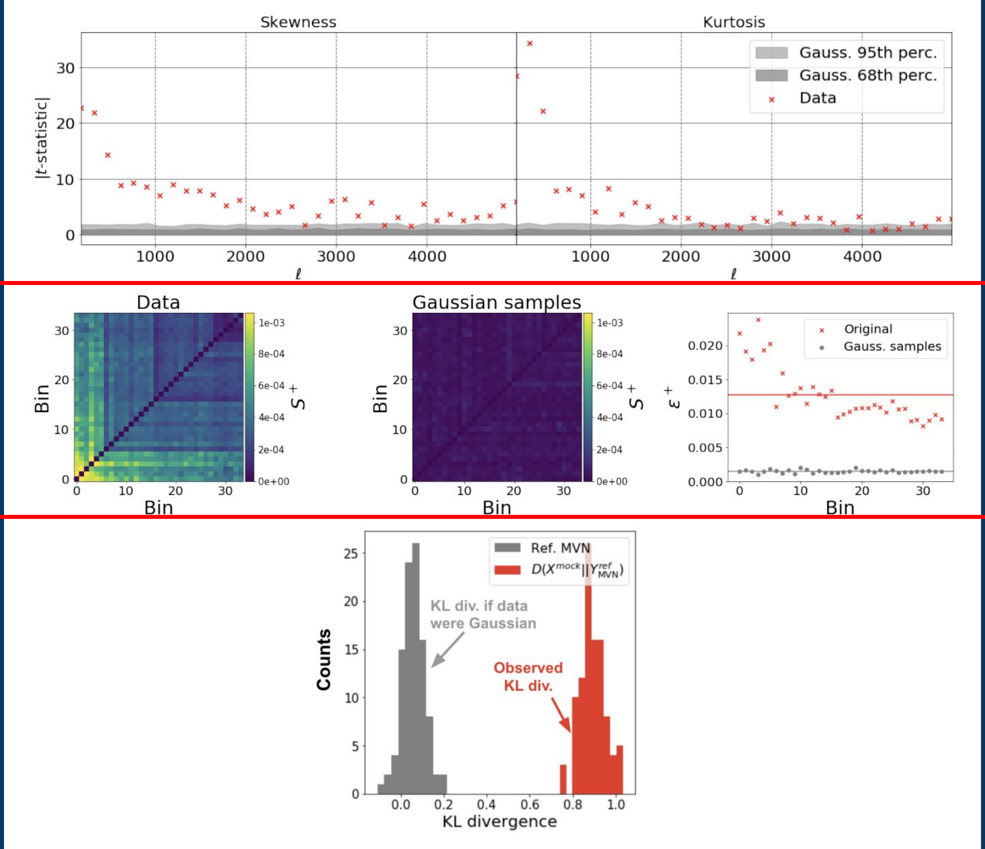


NG in the weak lensing convergence power spectrum

Test 1: t-stat of skewness and kurtosis

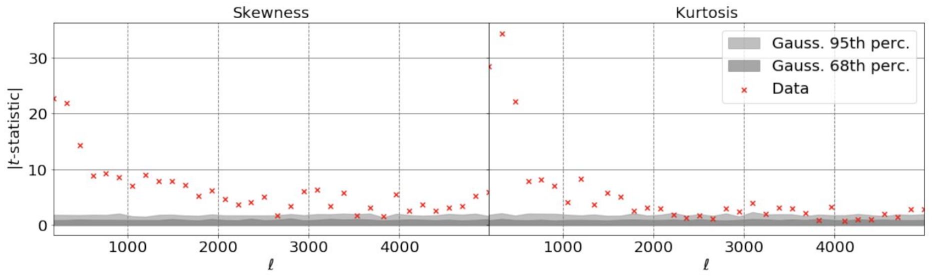
Test 2: transcovariance matrix

Test 3: KL divergence

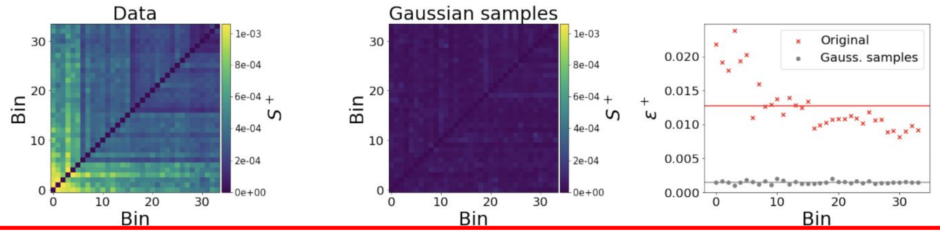


NG in the weak lensing convergence power spectrum

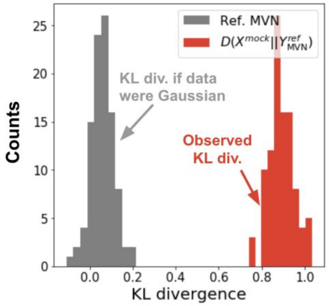
Test 1: t-stat of skewness and kurtosis



Test 2: transcovariance matrix

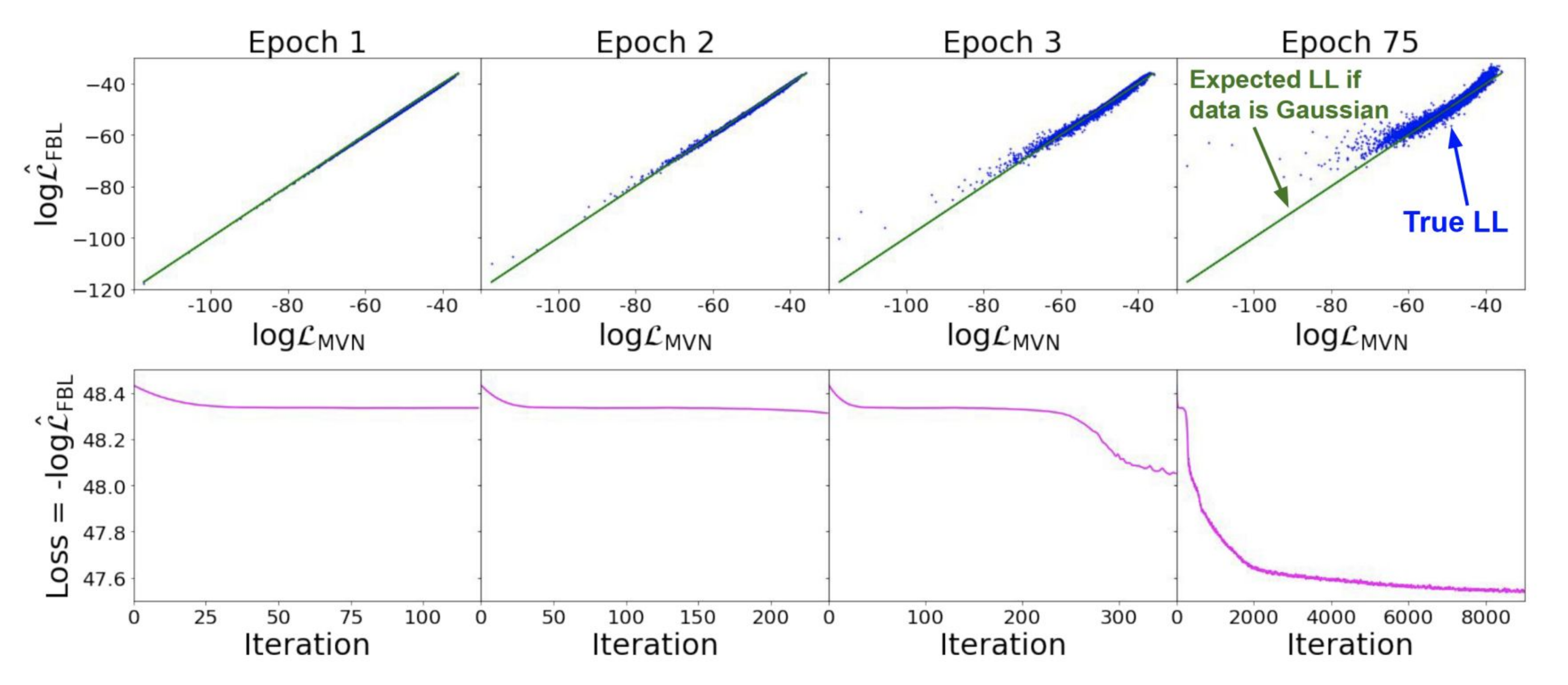


Test 3: KL divergence

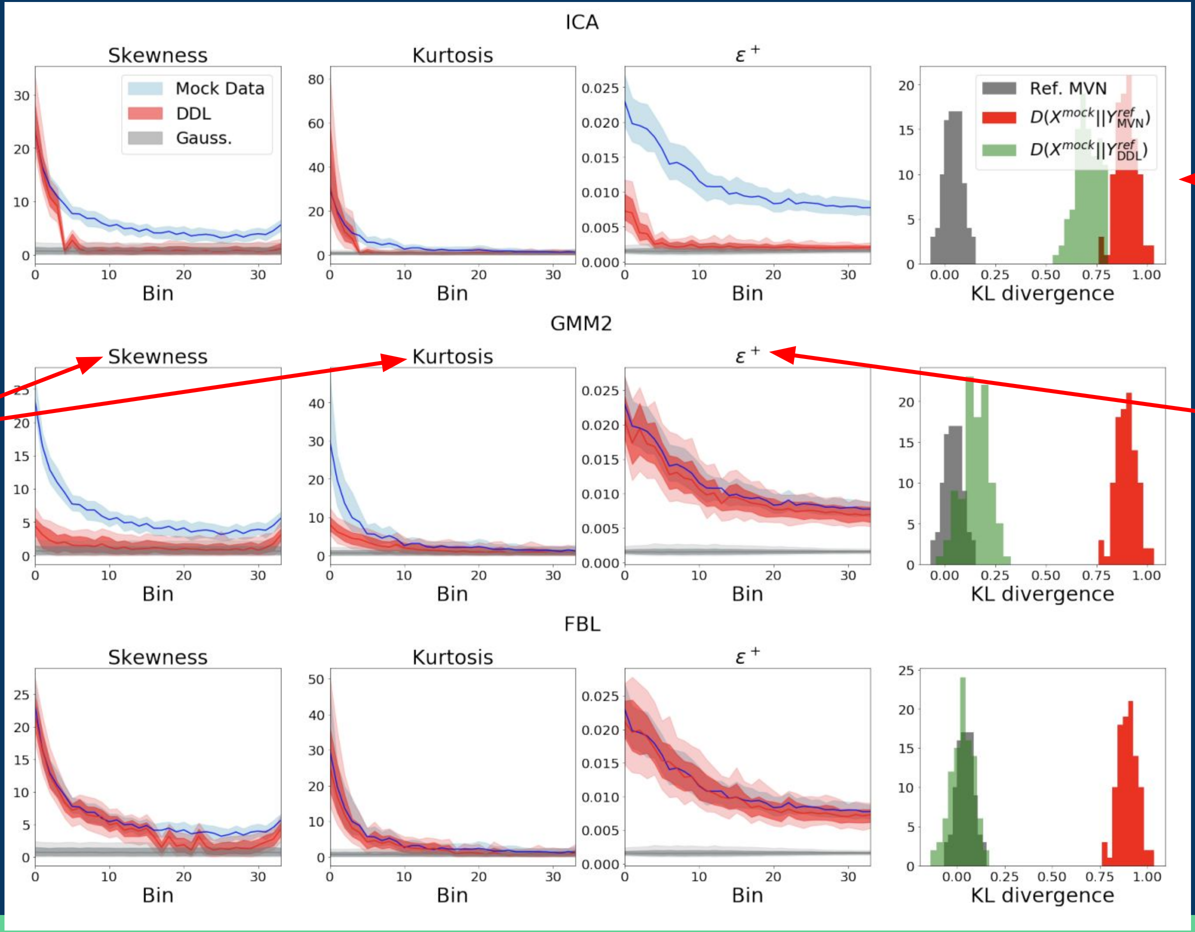


FBLs for the convergence power spectrum

Training the model and testing the likelihood



FBLs for the convergence power spectrum



Test 1: t-stat of skewness and kurtosis

Test 2: transcovariance matrix

Test 3: KL divergence

Conclusions

For our mock weak lensing data, GMM and ICA fail at capturing different NG, while the FBL does so much better:

- ICA can capture NG in individual bins but not **between** bins.
- GMM can capture NG between bins but not **individual** bins.
- FBL can capture both.

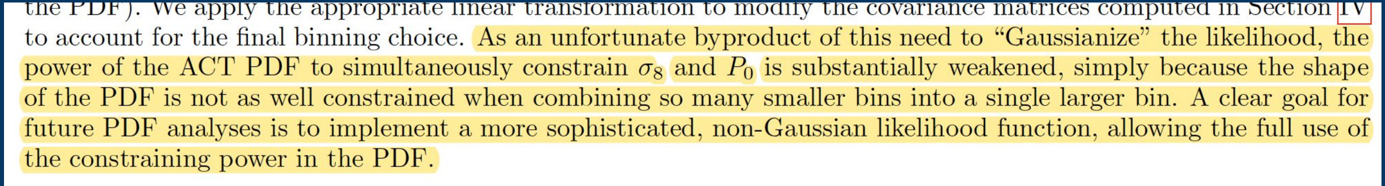
Data volume is not the only thing that determines the success/failure of a DDL: some understanding of the NG present in the data is crucial to select the right model.

Flexibility of FBLs can preclude them from a trial-and-error procedure that other DDLs can require.

Conclusions

WL in particular is interesting because:

- Seems to have some significant non-Gaussianities, even on scales where cosmic variance doesn't dominate (see also Sellentin+, 2018).
    - Some works Gaussianize the data (e.g. combining bins or with PCA) before inferring parameters, potentially destroying useful information, and conclude NG doesn't shift parameters (Lin+. 2019, Taylor+, 2019).



the PDF). We apply the appropriate linear transformation to modify the covariance matrices computed in Section IV to account for the final binning choice. As an unfortunate byproduct of this need to "Gaussianize" the likelihood, the power of the ACT PDF to simultaneously constrain σ_8 and P_0 is substantially weakened, simply because the shape of the PDF is not as well constrained when combining so many smaller bins into a single larger bin. A clear goal for future PDF analyses is to implement a more sophisticated, non-Gaussian likelihood function, allowing the full use of the constraining power in the PDF.

ACT Thermal SZ one-point PDF (Hill+, 2015)

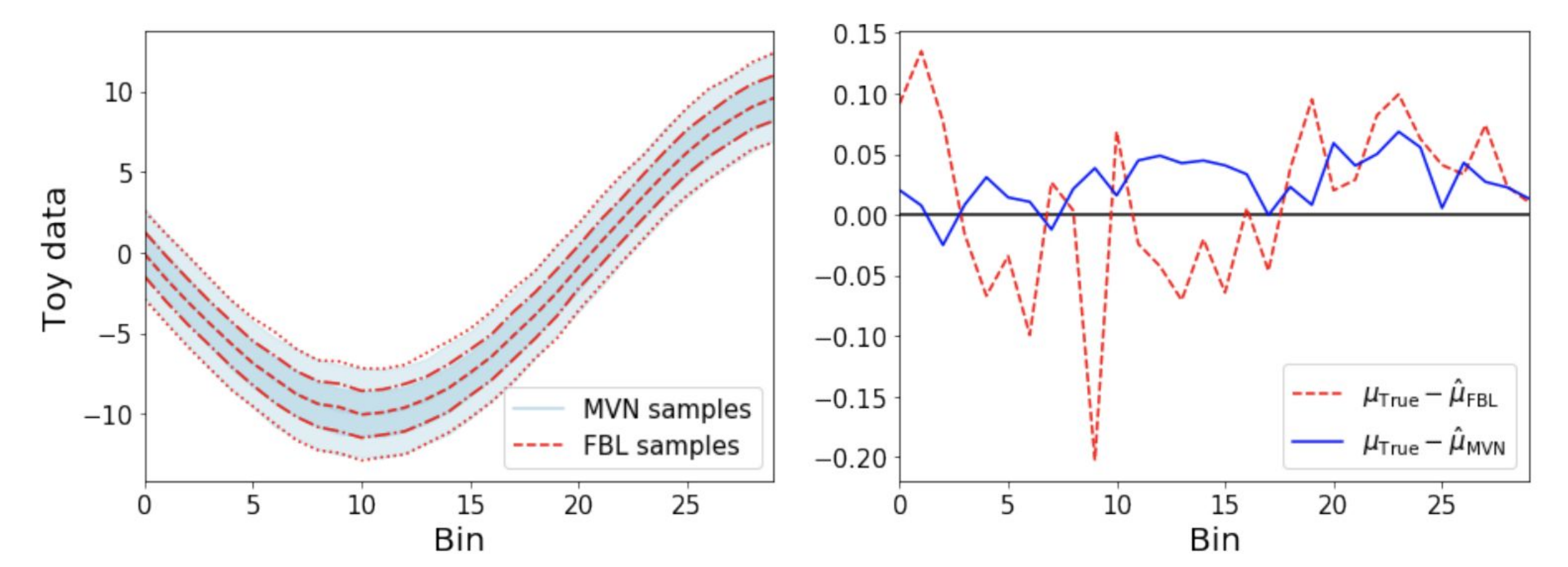
- Shortcomings of ICA in addressing pairwise non-Gaussian correlations in WL data:
    - works have used ICA dimensionality reduction before inferring parameters from weak lensing data and concluded NG don't impact parameter constraints considerably (Gupta+, 2018)

Questions?

arXiv: 2007.05535

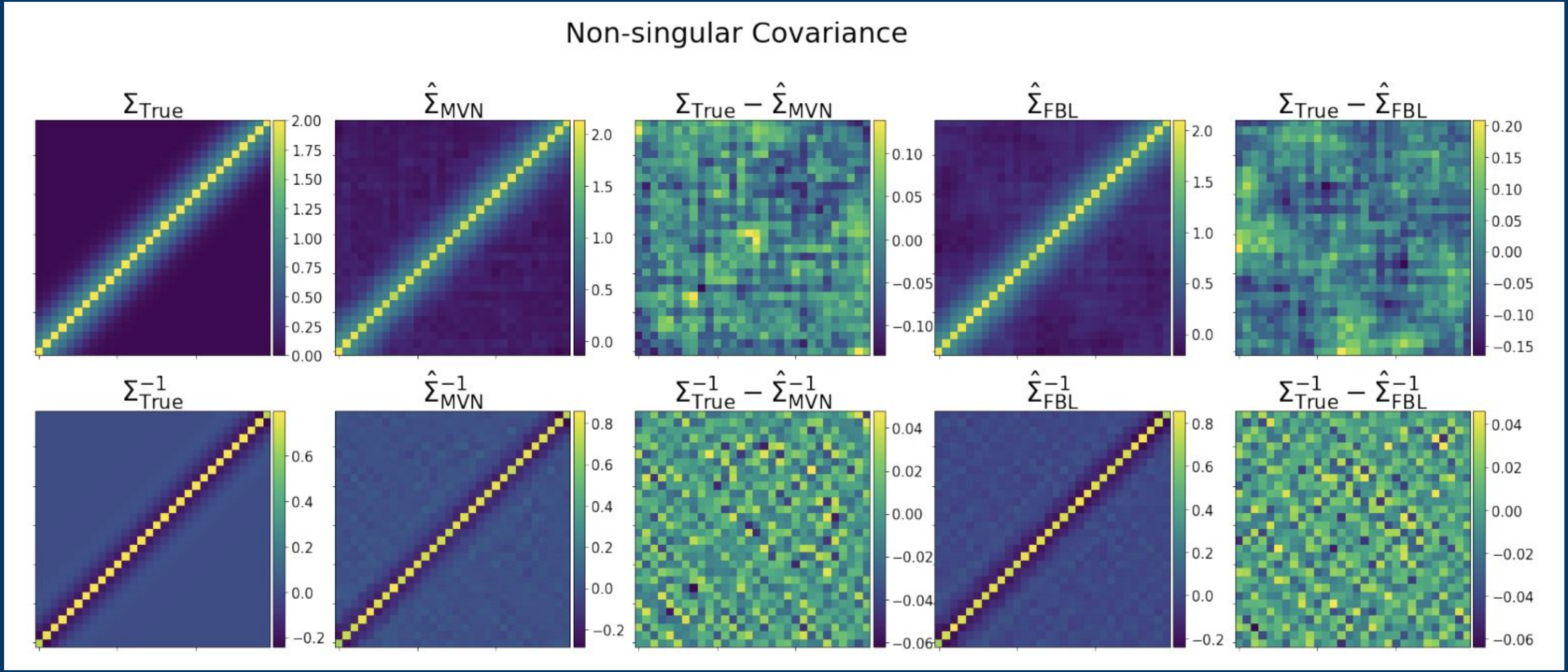
Samples vs likelihood quality

Non-singular covariance



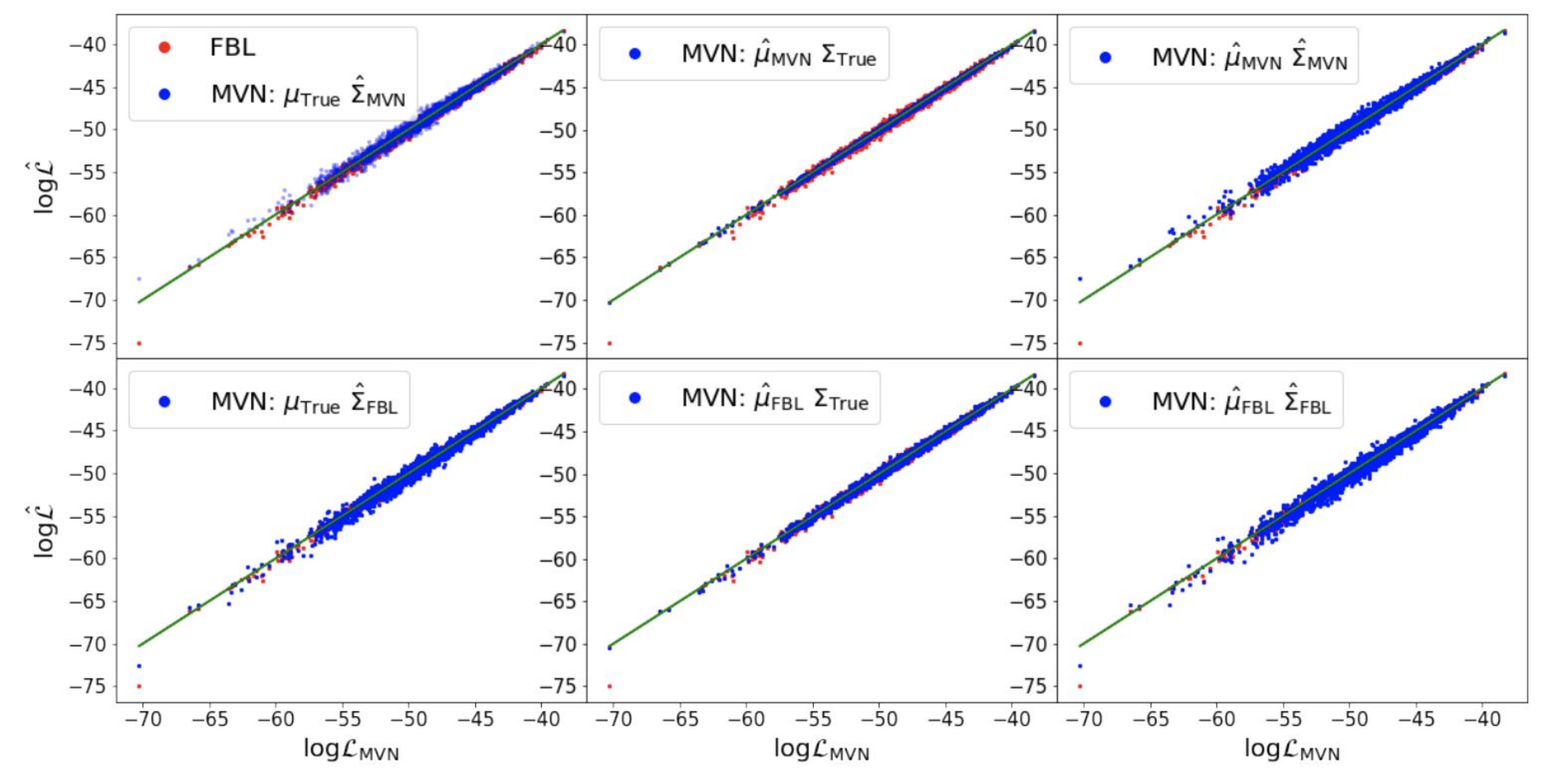
Samples vs likelihood quality

Non-singular covariance



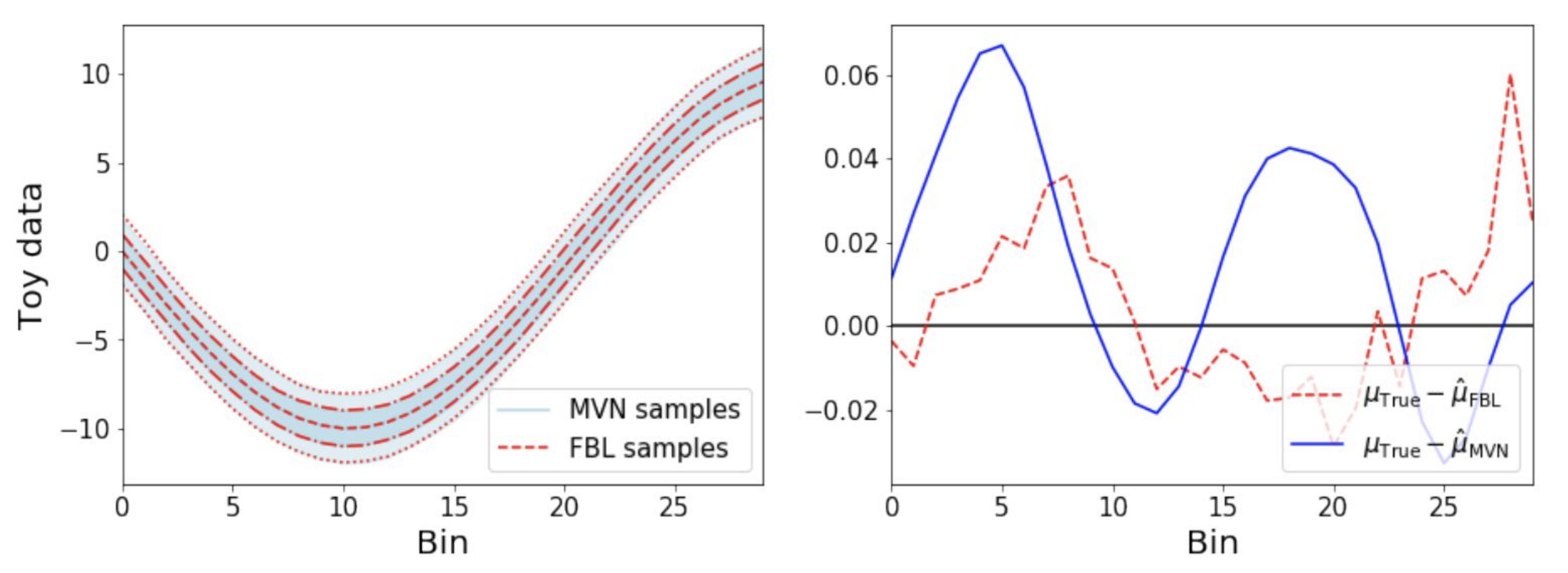
Samples vs likelihood quality

Non-singular covariance



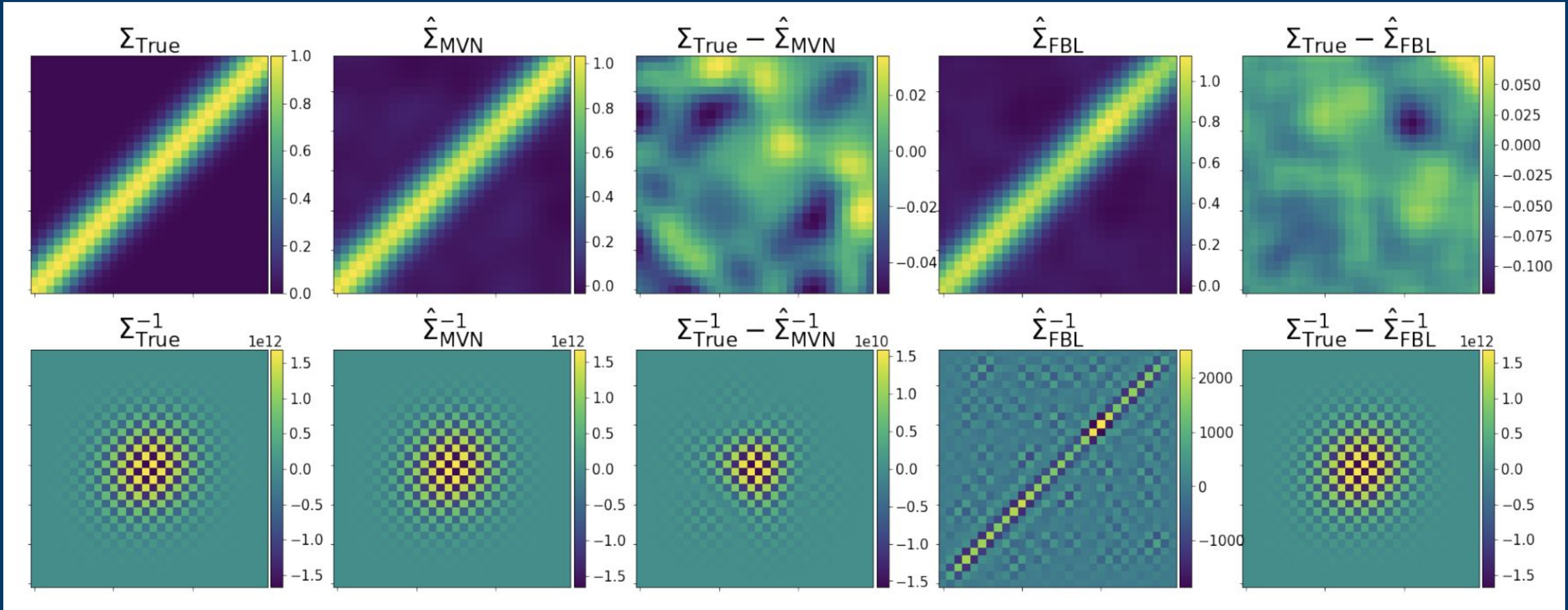
Samples vs likelihood quality

Singular covariance



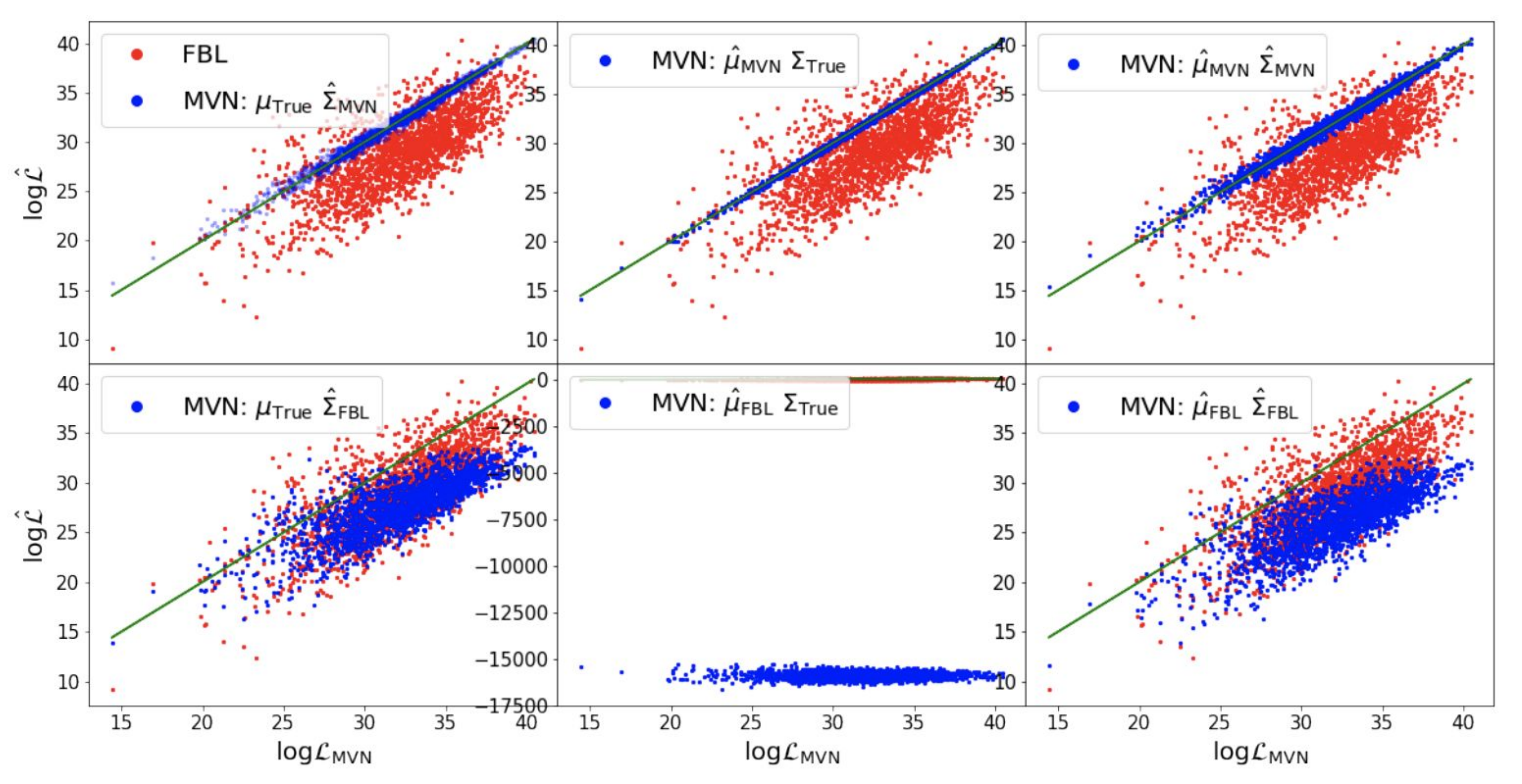
Samples vs likelihood quality

Singular covariance

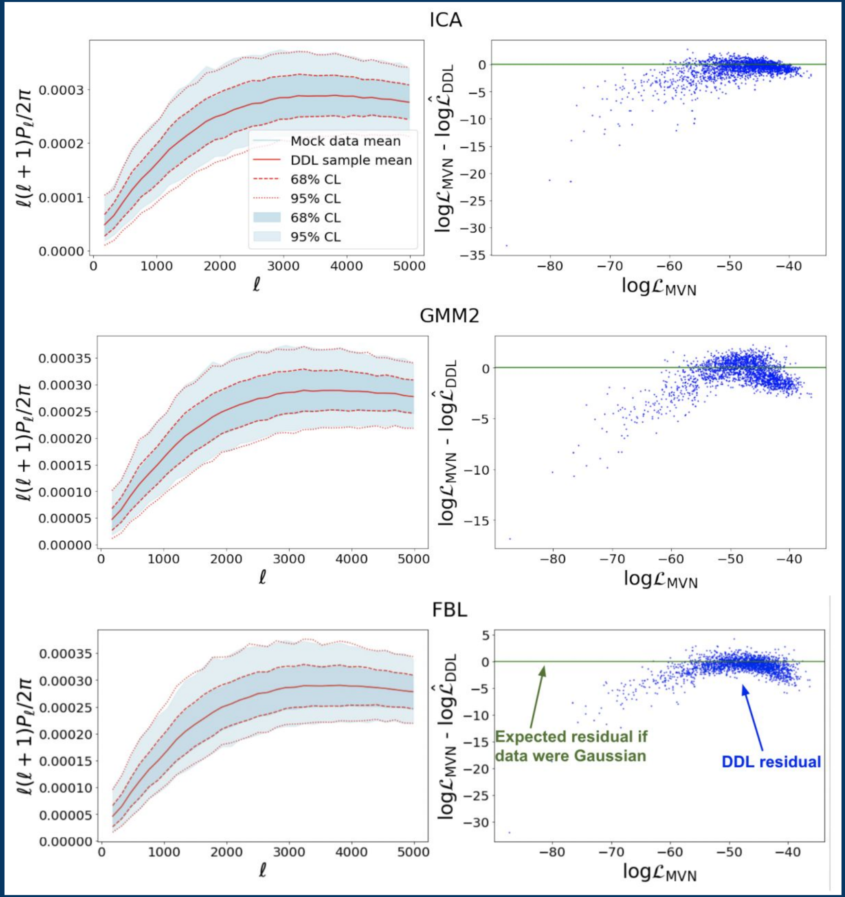


Samples vs likelihood quality

Singular covariance



Samples vs likelihood quality



Expected residual if data were Gaussian

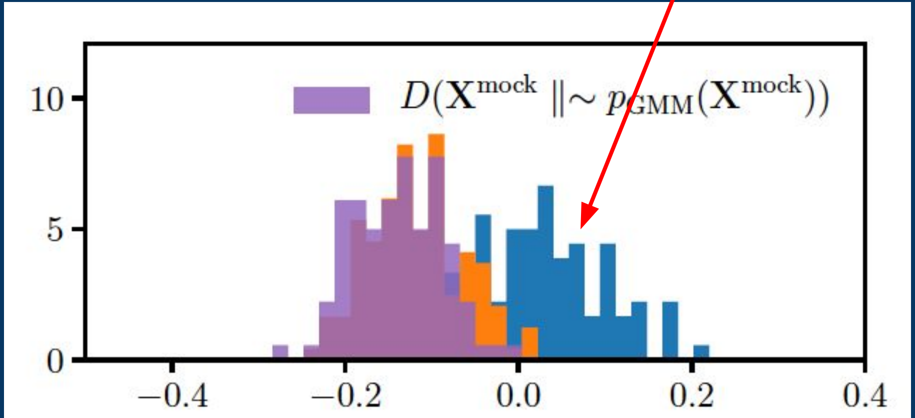
DDL residual

NG in the galaxy power spectrum

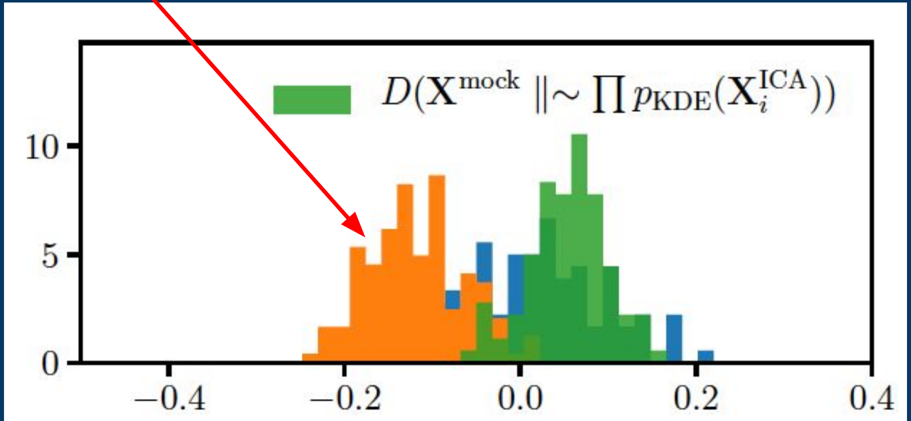
Hahn et al. (2018): BOSS 2,048 NGC galaxy power spectrum mocks. Find small posterior shifts doing importance sampling with ICA likelihood (< 0.5 σ).

Blue = gaussian

orange = data



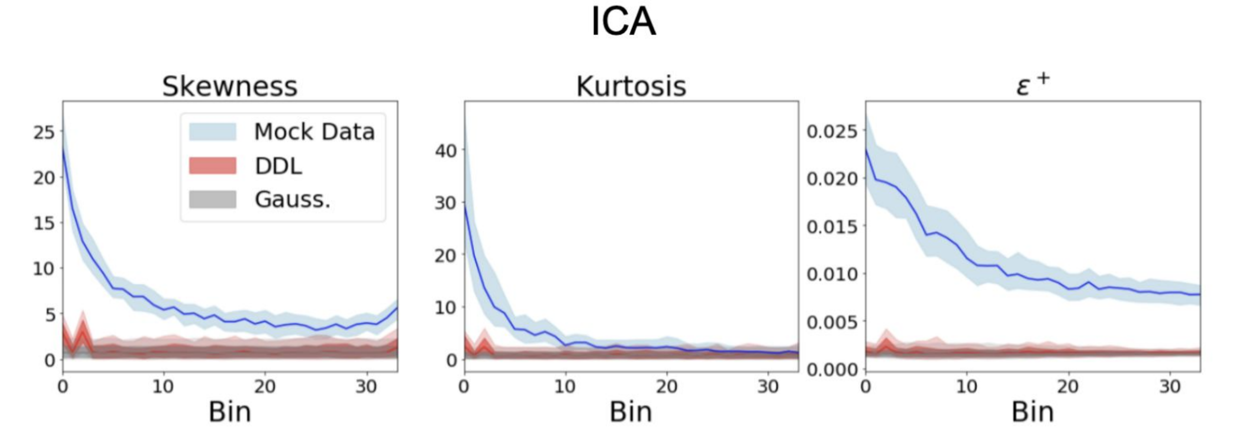
KL divergence

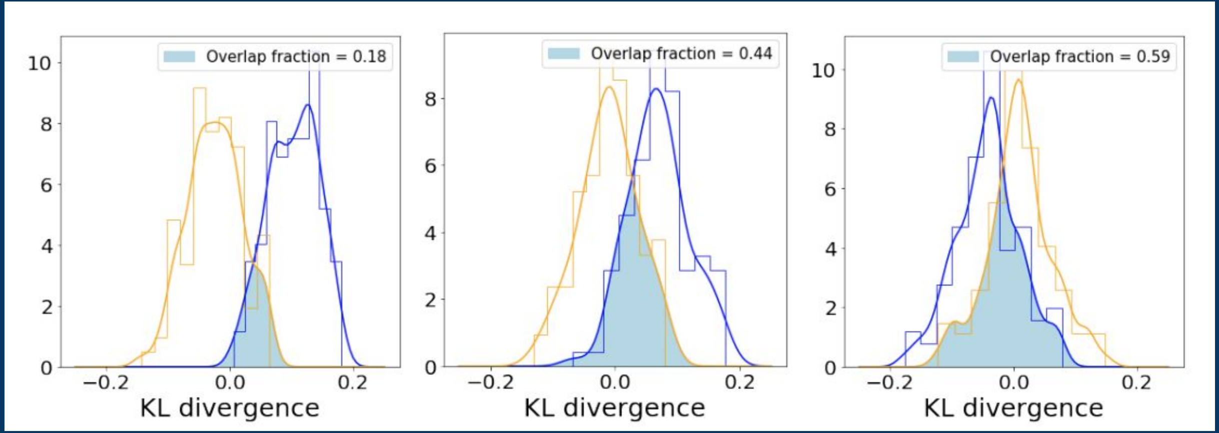


KL divergence

NG in the galaxy power spectrum

But...



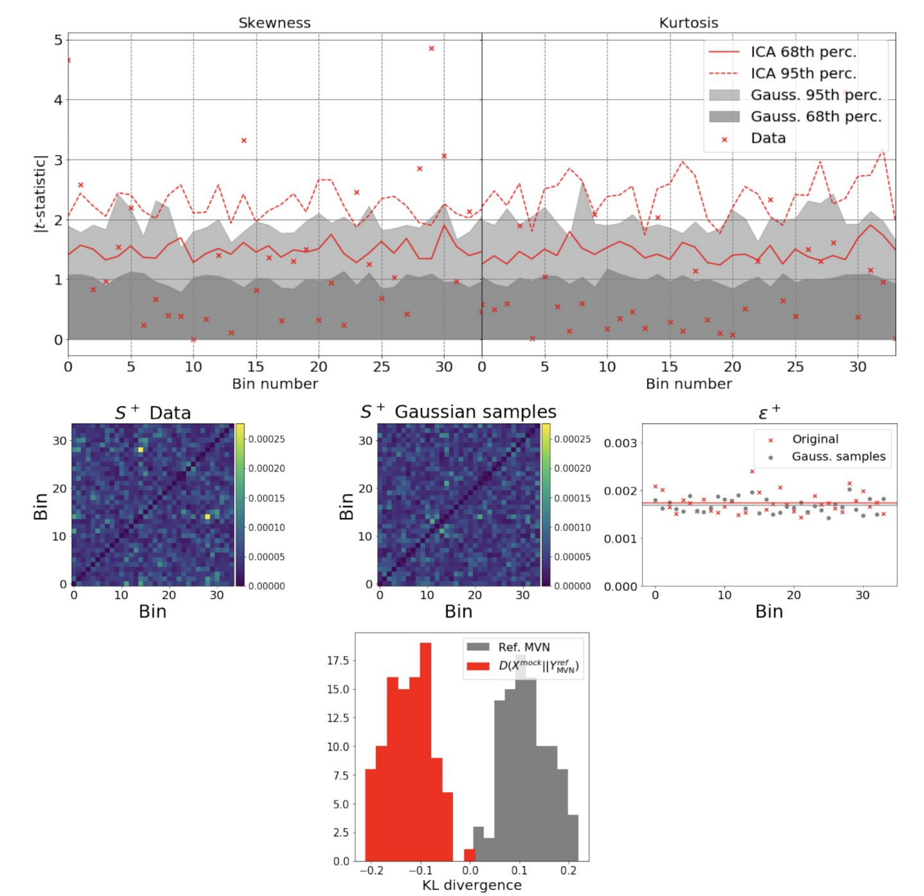


NG in the galaxy power spectrum

Test 1: t-stat of skewness and kurtosis

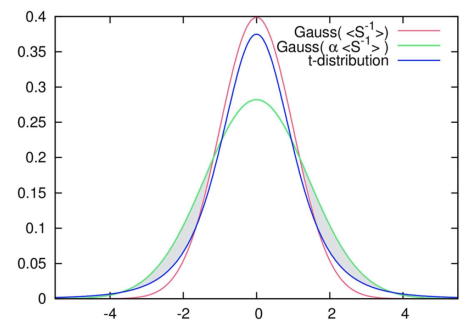
Test 2: transcovariance matrix

Test 3: KL divergence

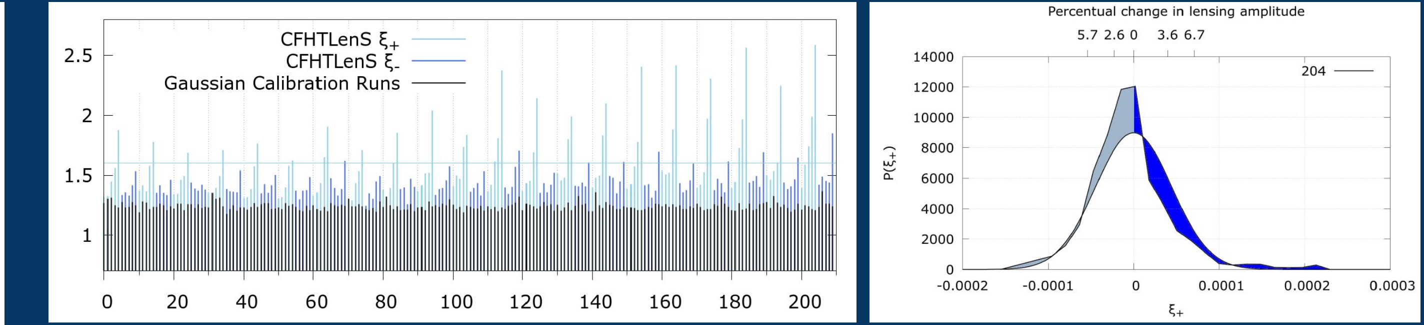


Gaussian likelihoods

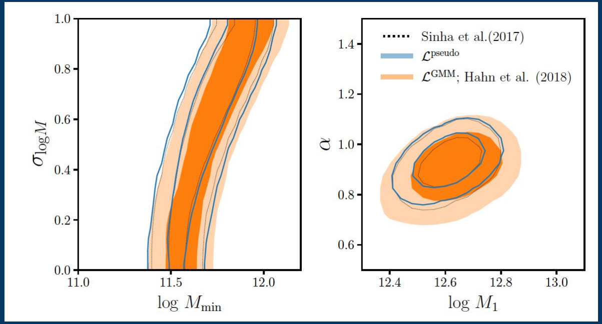
Estimating covariances (Sellentin+ 2016)



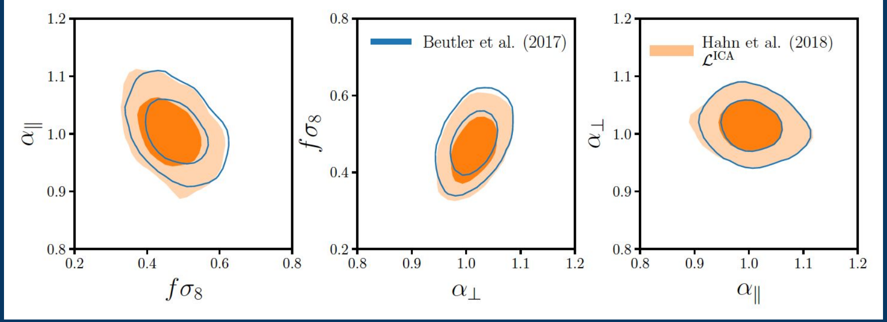
Weak lensing shear non-Gaussianity (Sellentin+, 2018)



Group multiplicity function (with GMM)



Galaxy power spectrum (with ICA)



Large-scale structure with non-Gaussian likelihoods (<0.5σ shifts, Chang+ 2018)