Title: Beyond Summary Statistics: Leveraging Generative Models for Robust and Optimal Field-Level Weak Lensing Analysis - VIRTUAL

Speakers: Biwei Dai

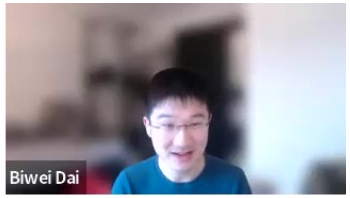
Series: Cosmology & Gravitation

Date: December 12, 2023 - 11:00 AM

URL: https://pirsa.org/23120049

Abstract: Deep learning (DL) methods have demonstrated great potential for extracting rich non-linear information from cosmological fields, a challenge that traditional summary statistics struggle to address. Most of these DL methods are discriminative models, i.e., they directly learn the posterior constraints of cosmological parameters. In this talk, I will make the argument that learning the field-level likelihood function using generative modeling approaches such as Normalizing Flows usually leads to more effective extraction of cosmological information. This approach also enables anomaly detection to improve the robustness of the analysis. To scale the modeling to high dimensional data and improve its generalization capabilities, we further incorporate physical inductive biases, such as symmetries and multiscale structure, into the architecture of the normalizing flow models. On mock weak lensing maps, I will show that the model leads to significant improvement in constraining power compared to power spectrum and alternative DL models. I will also show that it is able to detect domain shifts between training simulations and test data, such as noise miscalibration and baryonic effect, which, if left unaddressed, could introduce systematic biases in parameter constraints. Finally, I will talk about our ongoing work on applying this model to the field-level cosmic shear analysis for HSC.

Zoom link https://pitp.zoom.us/j/97478701784?pwd=UVg4TVp1WFArcXNETG5ITGd0S0NuZz09

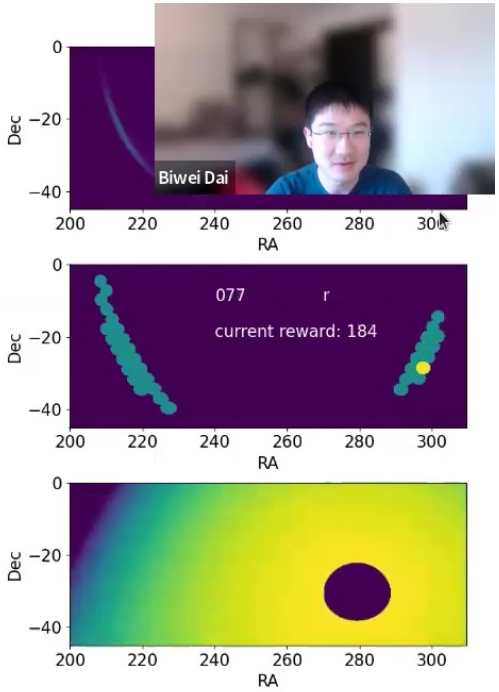


Beyond Summary Statistics: Leveraging Generative Models for Field-Level Weak Lensing Analysis

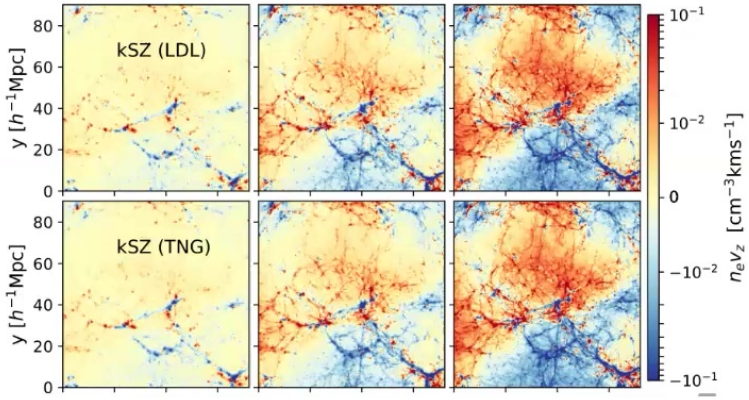
Biwei Dai
UC Berkeley
December 12 @ Perimeter Institute Cosmology Seminar

works with Uroš Seljak, Xiangchong Li, Rachel Mandelbaum, Divij Sharma, Francisco Villaescusa-Navarro

Astrophysics x Machine Learning



- Current works:
  - 80% of my time:
    - Generating mock weak lensing maps from ray-tracing simulations
    - Extracting cosmological information at the field level from WL maps
  - 20% of my time:
    - Kilonova follow-up observation with reinforcement learning
    - Learning representations of variable star light curves for novelty discovery

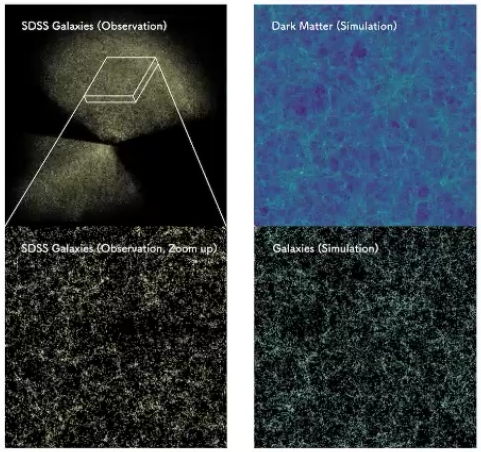


- Previous works:
  - Improve the accuracy of fast PM simulations
  - Model baryon observables from DMO simulations
  - Variational inference
  - Fast gravitational wave inference
  - Develop new ML algorithms (normalizing flows, samplers, etc.)

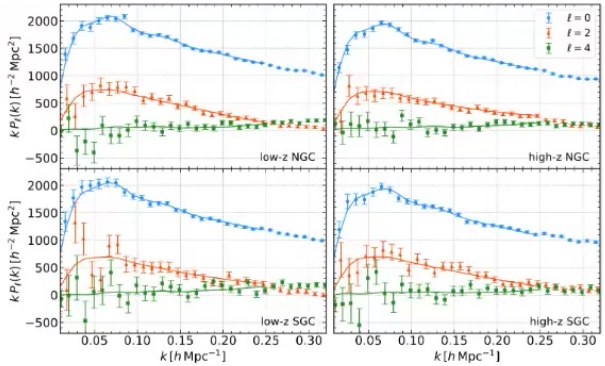
Cosmological analysis based on summary statist

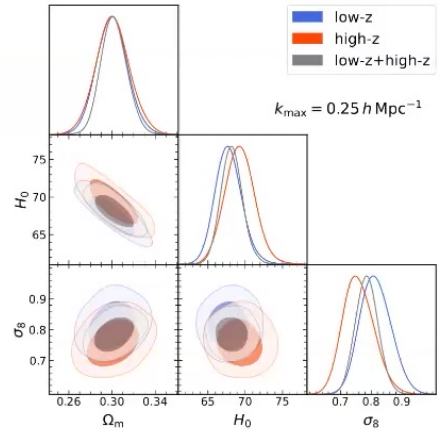
data x

cosmologic



summary statistics S





Compressing data x to summary statistics S

Gaussian likelihood on summary statistics S: p(S|y)

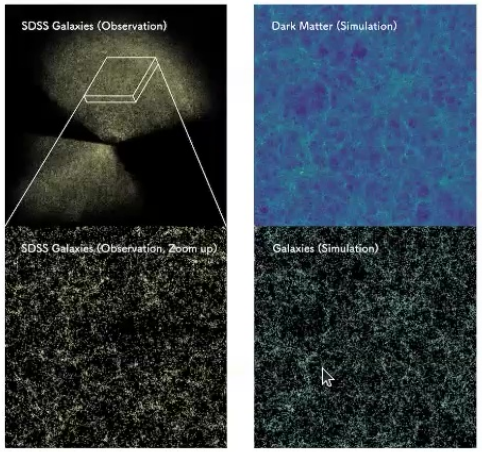
- ▷ Cosmological analysis based on two-point summary statistics: p(S|y)
  - ○ For non-gaussian data, usually leads to **information loss**

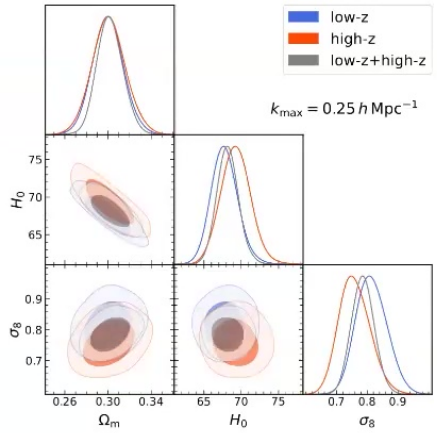
Credit: Takahiro Nishimichi, Kobayashi et al. 2022

Field-level cosmological inference

data x

cosmologic

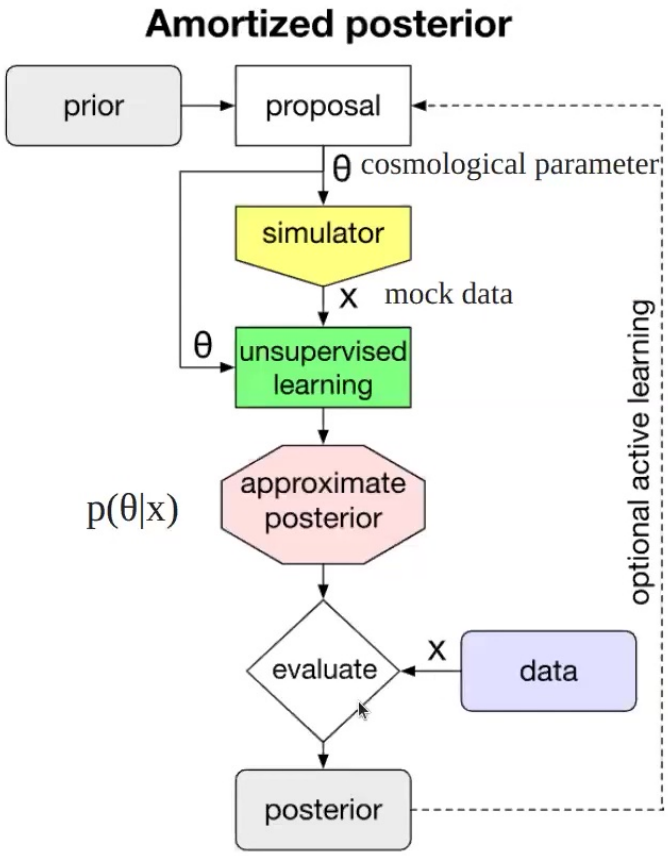


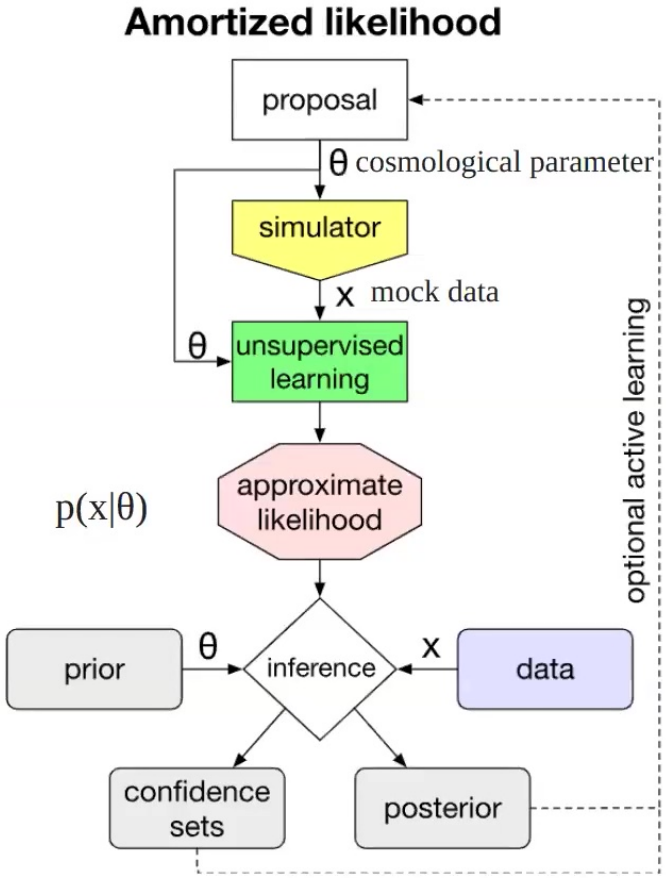


- Field-level inference
  - Pro: **No information loss** due to data compression.
  - Reconstruction of initial conditions with forward modeling and marginalize over ICs
  - Deep learning allows us to directly extract information at the field level (simulation-based inference)

Credit: Takahiro Nishimichi, Kobayashi et al. 2022

Simulation Based Inference (SBI)



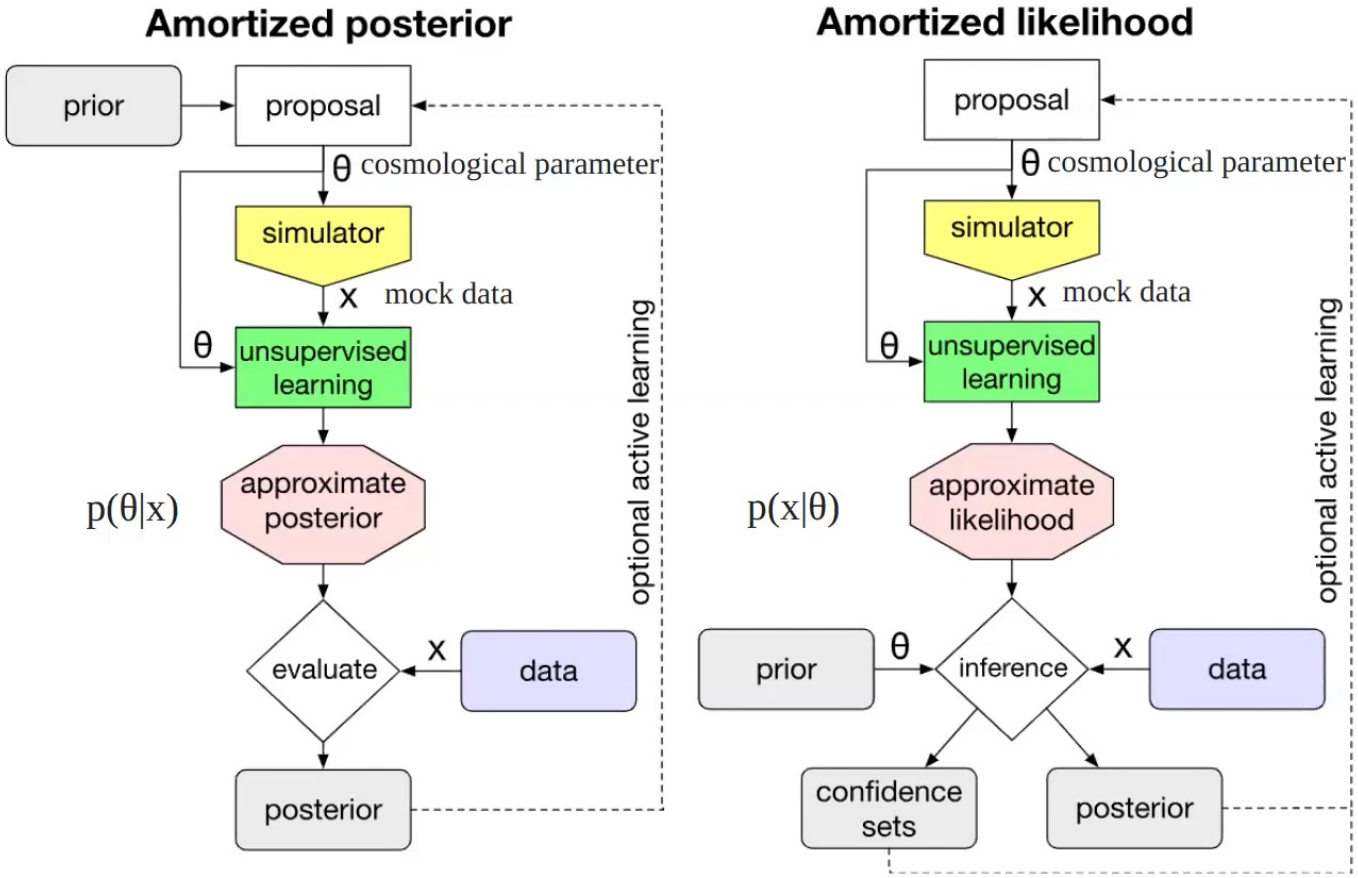


discriminative models

generative models

Cranmer et al. 2020

Simulation Based Inference (SBI)



discriminative models

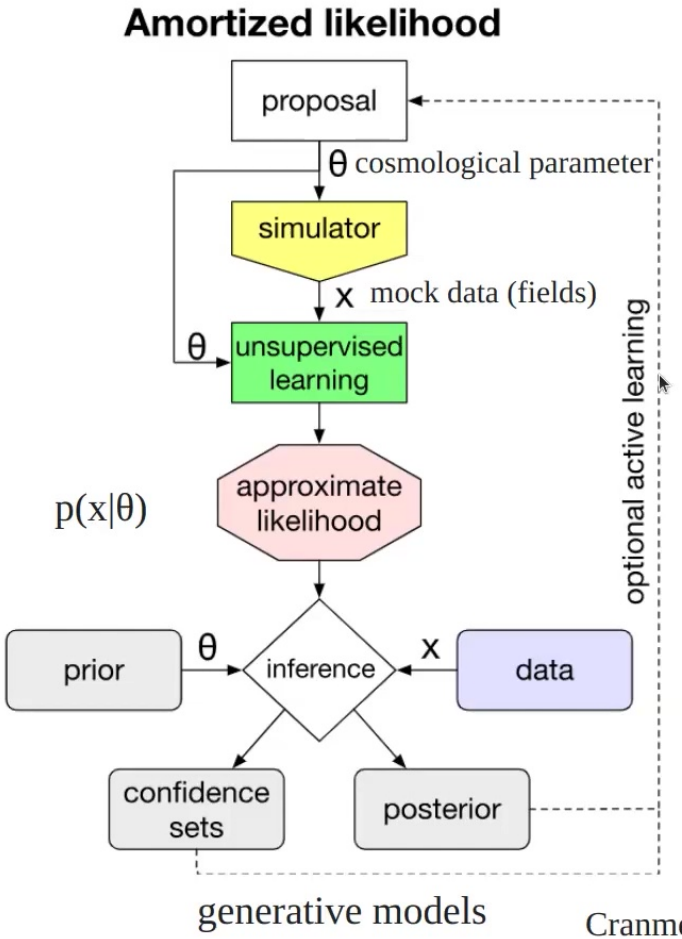
generative models

Cranmer et al. 2020

Neural networks are universal approximators.
Given a neural network with high enough capacity, and in the limit of infinite training data (x, θ), the learned likelihood/posterior will converge to the true solution.

End of story?

Simulation Based Inference (SBI) with generativ



Cranmer et al. 2020

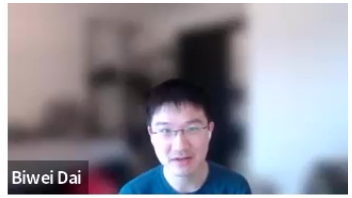
Potential issues of SBI:

1. **The simulations may not overlap with the reality**
2. **We don't understand how the constraints are made / where the information is coming from**
3. **Cosmological simulations are expensive**

SBI with generatives models:

- Field-level analysis: input data x is a 2D map, not summary statistics
- Generative models: learn the likelihood function, obtain the posterior with Bayes theorem

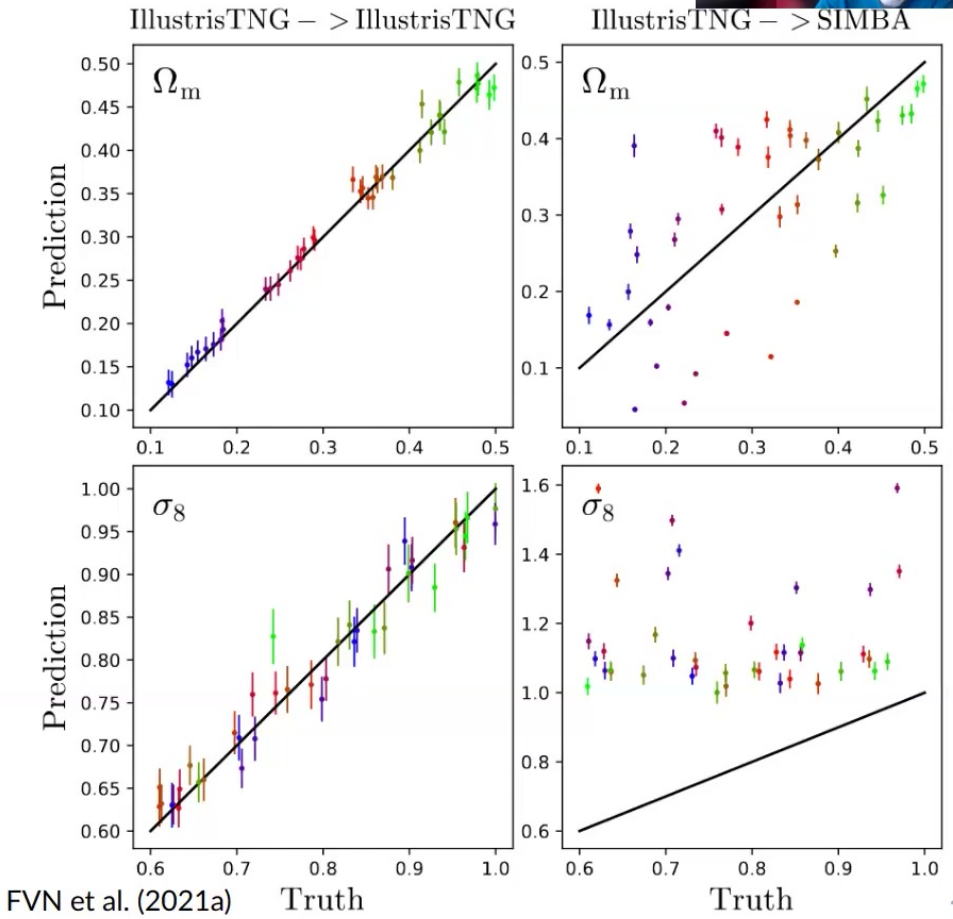
Why generative models?



- Reliable: verify the reliability of the analysis with goodness-of-fit test

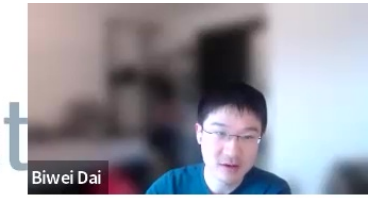
Robust cosmological data analysis

- How do we know we can trust our analysis?
- We need to assume that the simulations we trained on overlap with reality, but is this guaranteed?
- SBI models (CNN) trained on gas temperature maps of IllustrisTNG do not work on SIMBA.
- Marginalize over different baryon parameters, baryon models, N-body codes, etc.
- Tests to verify the reliability of the analysis



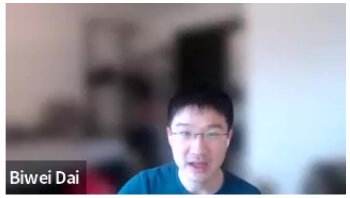
FVN et al. (2021a)

The analysis should pass various null tests / consist

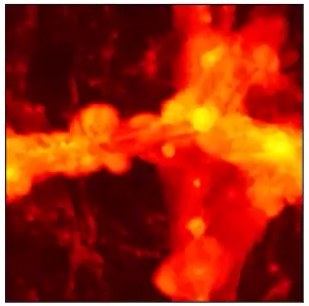


- Goodness-of-fit test with χ^2
  - The log-likelihood from generative models is a natural extension of the χ^2 statistic!
- Consistency test:
  - Robustness to different modeling choices
  - Internal consistency with different scales, redshift bins, different patches of the sky, etc.
- Null tests with different systematic effects
  - e.g., B mode tests in weak lensing

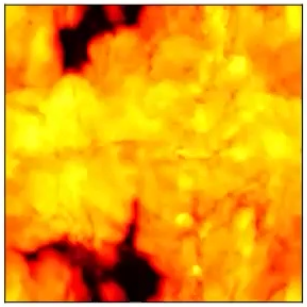
Goodness-of-fit test / Out-of-distribution detectio



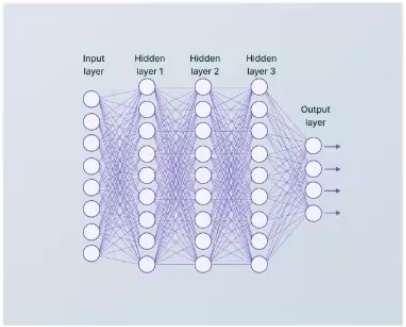
Training simulations

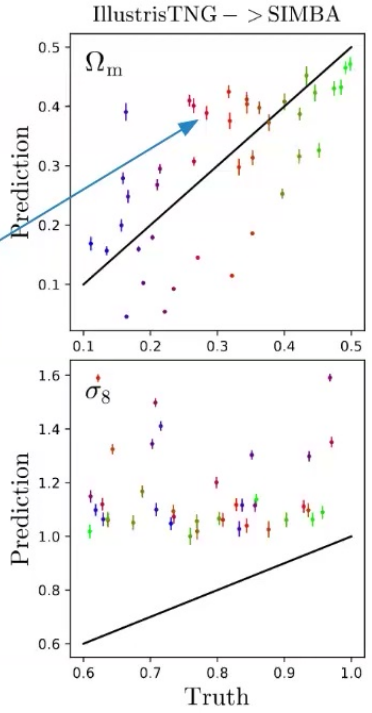


Test data / observation



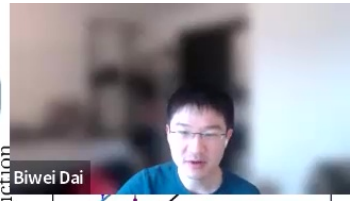
Discriminative models



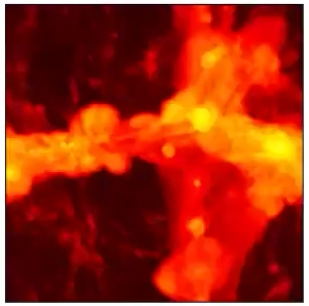


Biased parameter constraints due to distribution shifts, and we don't know it!

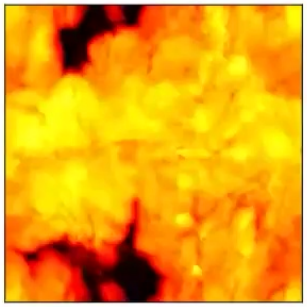
Goodness-of-fit test / Out-of-distribution detectio



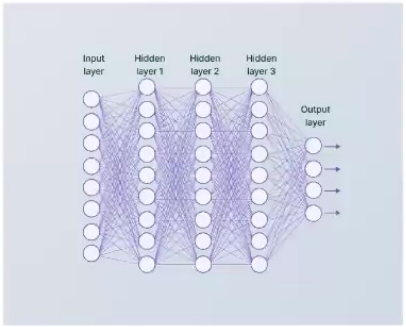
Training simulations



Test data / observation

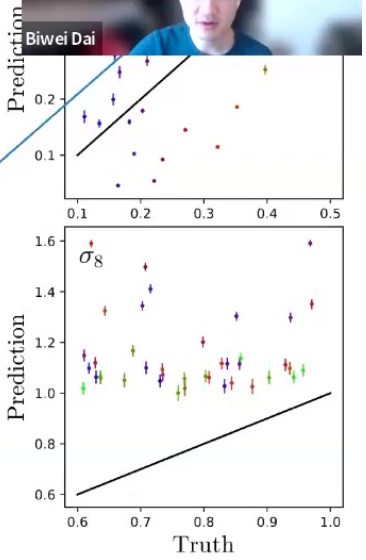


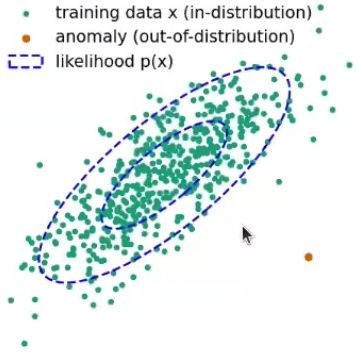
Generative models

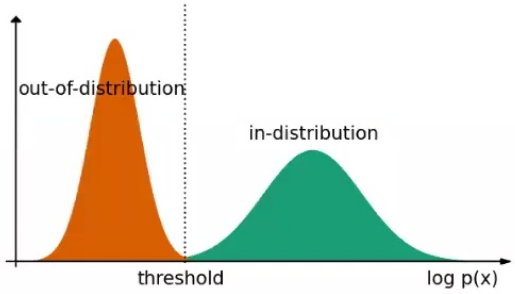


likelihood p(x|y)

MCMC

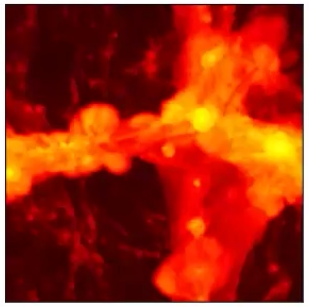




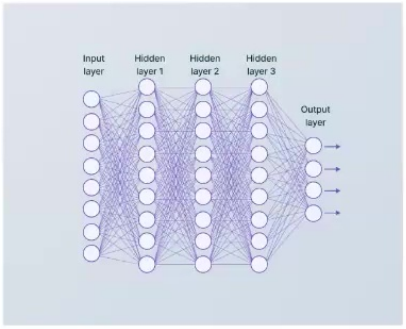


Goodness-of-fit test / Out-of-distribution detectio

Training simulations

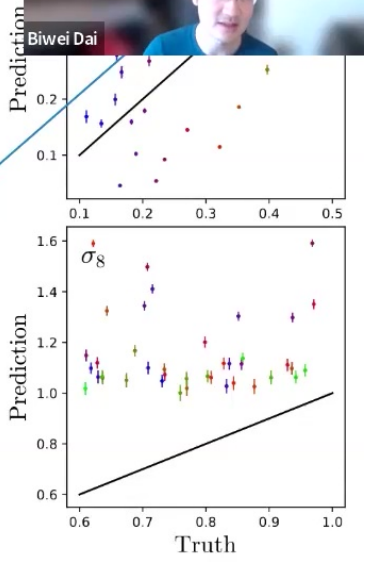


Generative models

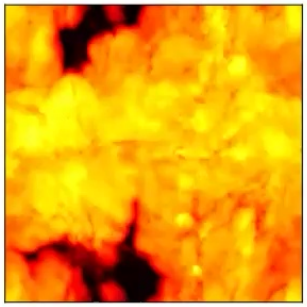


MCMC

likelihood p(x|y)

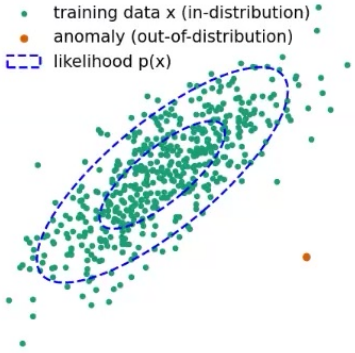


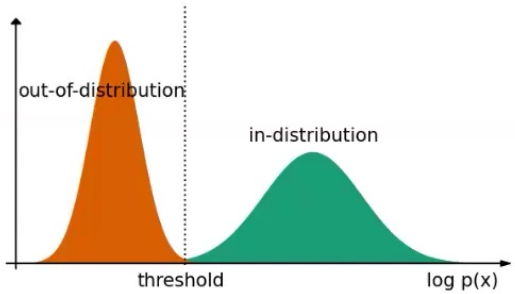
Test data / observation



The test data / observation doesn't look like training data, so we shouldn't trust our analysis!

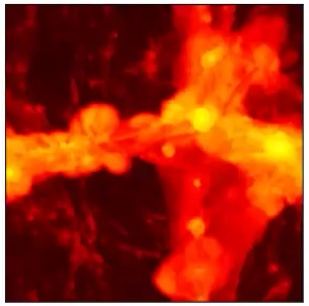
More conservative scale cuts, or improve the modeling of training simulations.



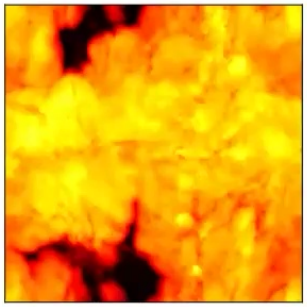


Goodness-of-fit test / Out-of-distribution detectio

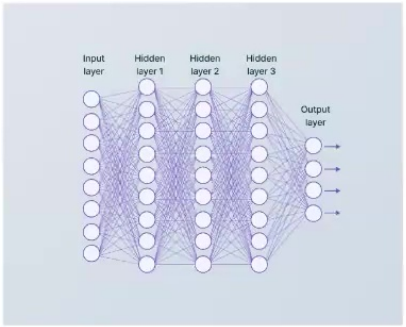
Training simulations



Test data / observation



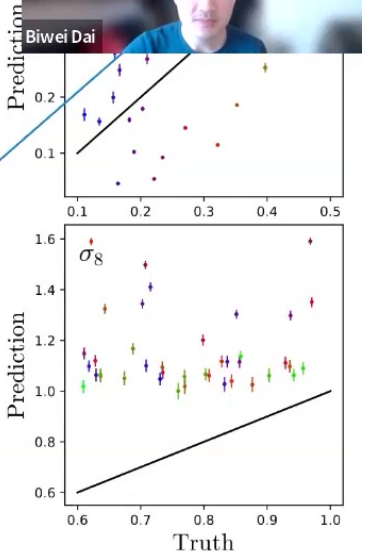
Generative models



Advantage 1: Generative models enable goodness-of-fit test to improve the robustness of analysis.

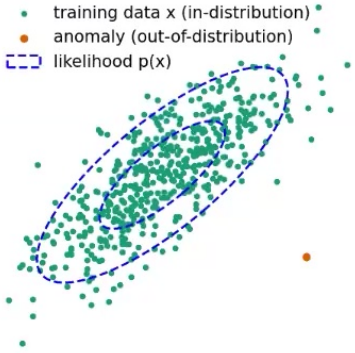
MCMC

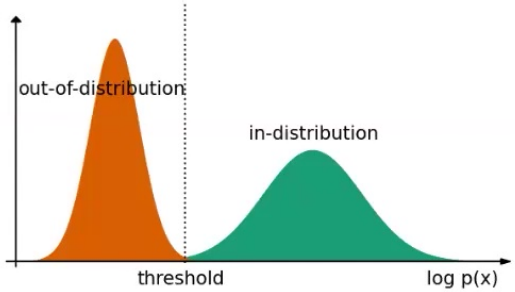
likelihood p(x|y)



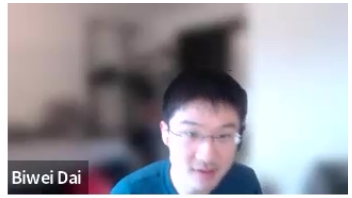
The test data / observation doesn't look like training data, so we shouldn't trust our analysis!

More conservative scale cuts, or improve the modeling of training simulations.





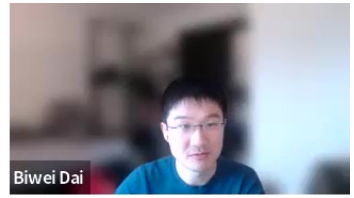
Why generative models?



- Reliable: verify the reliability of the analysis with goodness-of-fit test
- Interpretable: understand where the information is coming from

"Where is the extra information coming from?"

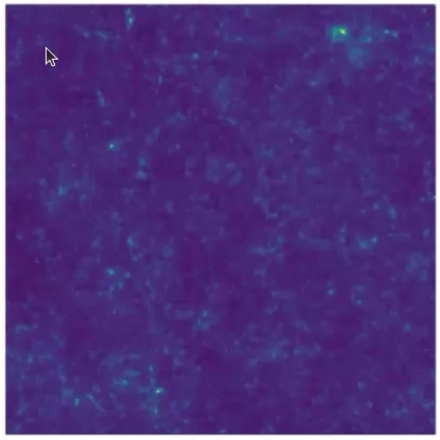
"You need to show why the other cosmological models are ruled out"



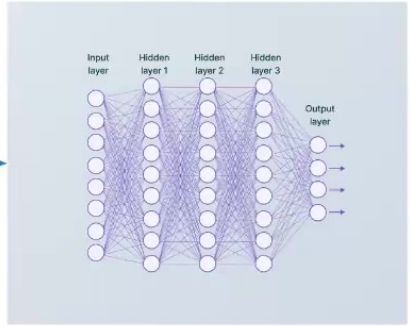
"Where is the extra information coming from?"

"You need to show why the other cosmological models are ruled out"

Input WL map

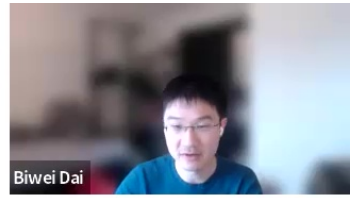


Generative models



MCMC

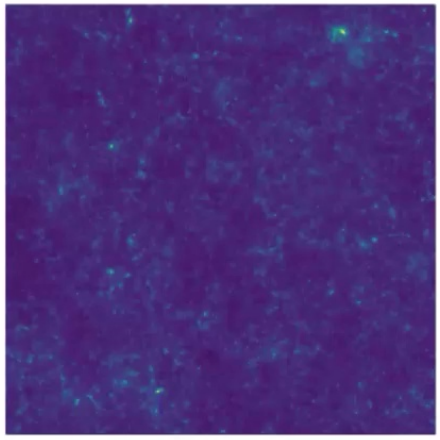
$\sigma_8 = 0.76 \pm 0.02$



"Where is the extra information coming from?"

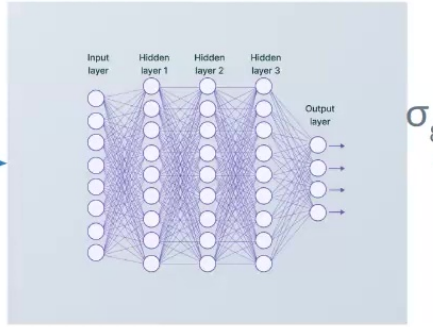
"You need to show why the other cosmological models are ruled out"

Input WL map



These are real maps generated by our model!

Generative models

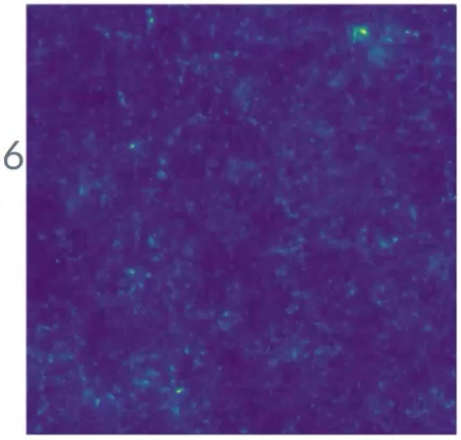


MCMC

$\sigma_8 = 0.76 \pm 0.02$

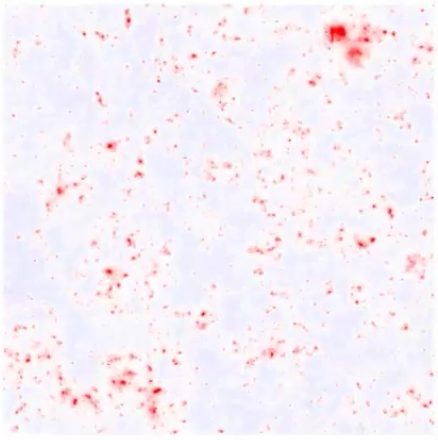
$\sigma_8 = 0.816$

Generated sample



The same realization (latent code) as the input map, but assuming a different cosmology

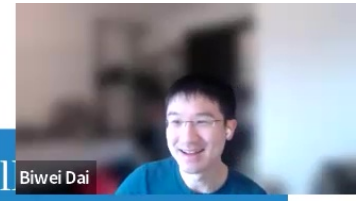
Difference



Generated sample - input map

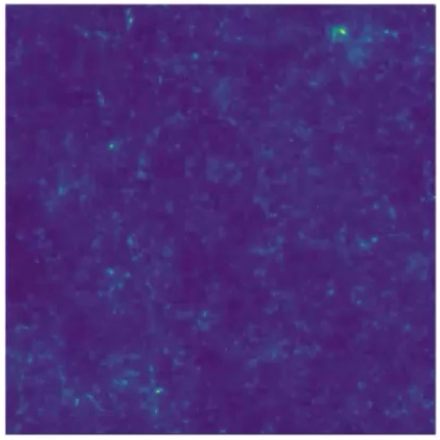
"Where is the extra information coming from?"

"You need to show why the other cosmological models are ruled out"

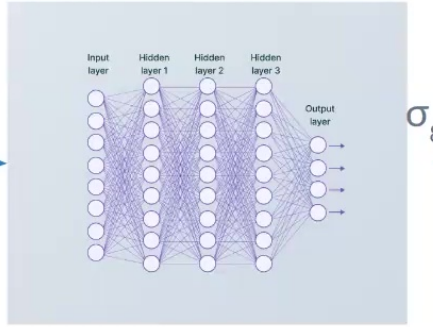


My model tells us that halos from high σ_8 cosmology are too massive!

Input WL map

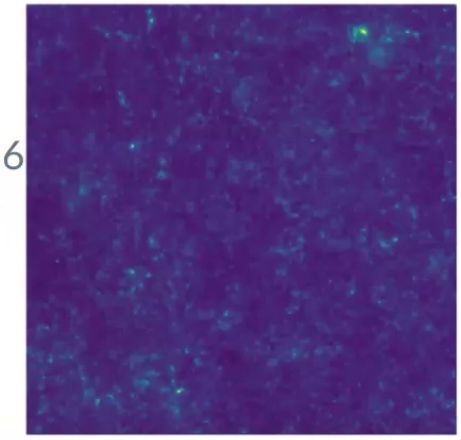


Generative models

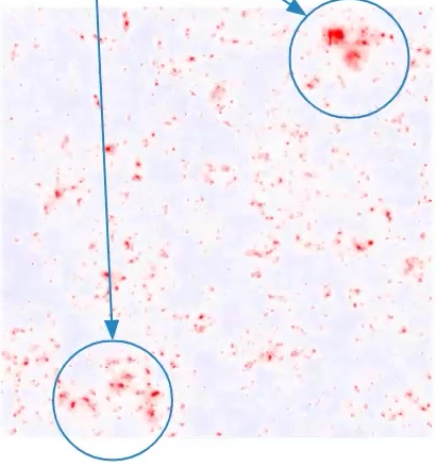


$\sigma_8 = 0.816$

Generated sample



Difference



MCMC

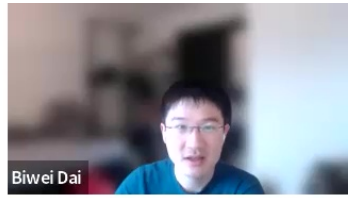
$\sigma_8 = 0.76 \pm 0.02$

These are real maps generated by our model!

The same realization (latent code) as the input map, but assuming a different cosmology

Generated sample - input map

Discriminative v.s. Generative Models



	Discriminative Models	Generative Models
Target / Model Output	Posterior p(y|x)	Likelihood p(x|y) (Use Bayes rule to calculate posterior.)
Example	Train NNs to predict y and its uncertainty	This talk: train normalizing flows to learn field-level likelihood p(x|y)
Asymptotic Error		
Sample Complexity		

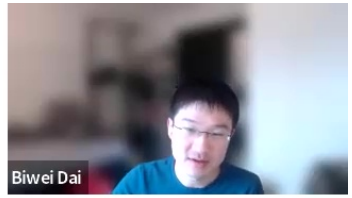
Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*.
Zheng et al. 2023. Revisiting Discriminative vs. Generative Classifiers: Theory and Implications. *International conference on machine learning* (pp. 42420-42477). PMLR

Discriminative v.s. Generative Models

	Discriminative Models	Generative Models
Target / Model Output	Posterior p(y|x)	Likelihood p(x|y) (Use Bayes rule to calculate posterior.)
Example	Train NNs to predict y and its uncertainty	This talk: train normalizing flows to learn field-level likelihood p(x|y)
Asymptotic Error	**Low**	High
Sample Complexity	High O(n) for linear model	**Low** **O(logn) for linear model**

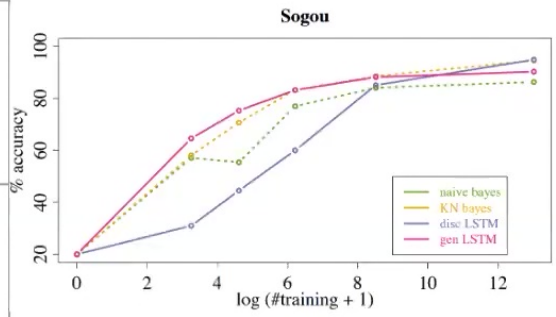
Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*.
Zheng et al. 2023. Revisiting Discriminative vs. Generative Classifiers: Theory and Implications. *International conference on machine learning* (pp. 42420-42477). PMLR

Discriminative v.s. Generative Models



	Discriminative Models	Generative Models
Target / Model Output	Posterior p(y|x)	Likelihood p(x|y) (Use Bayes rule to calculate posterior.)
Example	Train NNs to predict y and its uncertainty	This talk: train normalizing flows to learn field-level likelihood p(x|y)
Asymptotic Error	**Low**	High
Sample Complexity	High O(n) for linear model	**Low** **O(logn) for linear model**

"Two regimes" phenomenon

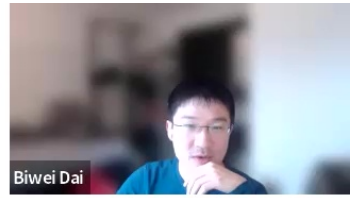


Yogatama et al. 2017 (https://arxiv.org/pdf/1703.01898v2.pdf)

Important for large n: field-level inference

Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*.
Zheng et al. 2023. Revisiting Discriminative vs. Generative Classifiers: Theory and Implications. *International conference on machine learning* (pp. 42420-42477). PMLR

Experimental evidence



- Figure of merit (constraining power): the reciprocal of the 1σ confidence area on the (Ω_m, σ_8) plane on WL maps

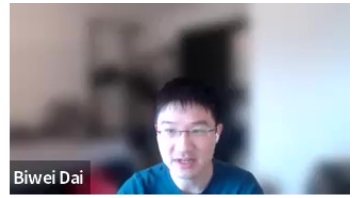
Method	n_g = 10arcmin^{-2}	n_g = 30arcmin^{-2}	n_g = 100arcmin^{-2}
Multiscale Flow	**95**	**246**	**733**
power spectrum	30 (30)	52 (51)	81 (79)
peak count	(40)	(85)	(137)
CNN	64(76)	146(129)	327(317)
scattering transform $s_0 + s_1 + s_2$	($\lesssim$ 50)	($\lesssim$ 140)	($\lesssim$ 329)

Generative model

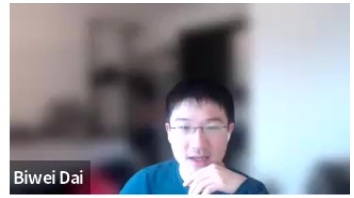
Discriminative model

(Advantage 3: Generative models are likely to extract more information (with a limited number of training simulations)!)

Why generative models?



- Reliable: verify the reliability of the analysis with goodness-of-fit test
- Interpretable: understand where the information is coming from
- (Efficient learning: train the model with fewer simulations)

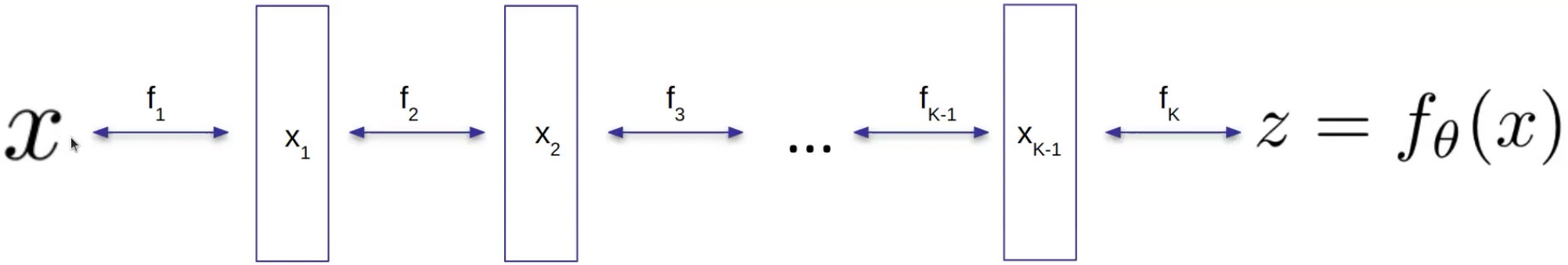


How do we model the high-dimensional field-level likelihood function p(x|y)?

- Use normalizing Flows to parametrize the probability distribution function

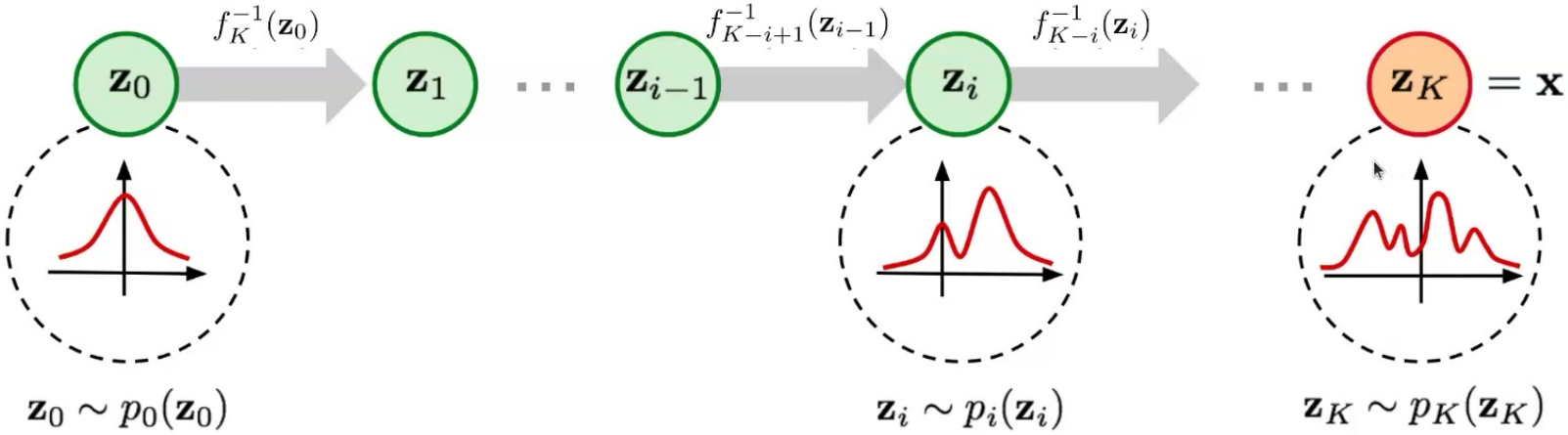
- Add physical inductive bias to scale to high-dimensions and improve generalization

Normalizing Flows



- Bijective mapping f between data x and latent variable z ($z = f(x)$, $z \sim \pi(z)$)
  - **Evaluate density**: $p(x) = \pi(f(x)) \, |\det(df/dx)|$
  - **Sample**: $x = f^{-1}(z)$ ($z \sim \pi(z)$)

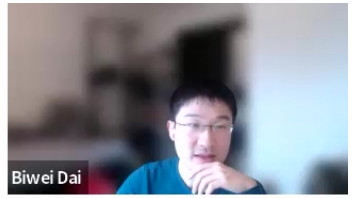
Normalizing Flows



Credit: https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

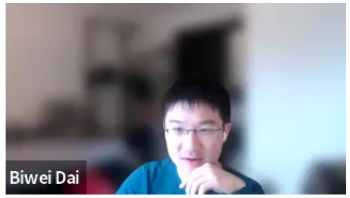
- Bijective mapping f between data x and latent variable z (z = f(x), z ~ π(z))
  - **Evaluate density**: p(x) = π(f(x)) |det(df/dx)|
  - **Sample**: x = f^{-1}(z) (z ~ π(z))

From Gaussian Random Fields, Lognormal Fields to



	NF transformation	Probability distribution function
Gaussian Random Fields	$\delta_{\rm GRF}(k) = \sqrt{P(k)}\delta_z(k)$	$p(\delta_{\rm GRF}) = \frac{1}{\sqrt{(2\pi)^n \prod_i P(k_i)}} \exp(-\sum_i \frac{\lvert\delta_{\rm GRF}\rvert^2}{2P(k_i)})$

From Gaussian Random Fields, Lognormal Fields to

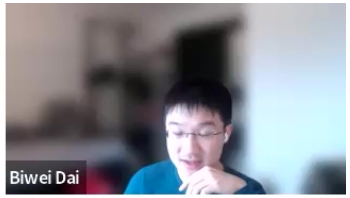


	NF transformation	Probability distribution function
Gaussian Random Fields	$\delta_{\rm GRF}(k) = \sqrt{P(k)}\delta_z(k)$	$p(\delta_{\rm GRF}) = \frac{1}{\sqrt{(2\pi)^n \prod_i P(k_i)}} \exp(-\sum_i \frac{\lvert\delta_{\rm GRF}\rvert^2}{2P(k_i)})$
Lognormal Fields	$\delta_{\rm LN} = \exp(\delta_{\rm GRF}) - C$	$p(\delta_{\rm LN}) = p(\delta_{\rm GRF})\lvert\frac{\partial \delta_{\rm GRF}}{\partial \delta_{\rm LN}}\rvert$
Generalization	$f_i(x) = \Psi_i(\mathcal{F}^{-1} T_i(k) \mathcal{F} x)$	$p(f_i(x)) = p(x) \prod_r \lvert\Psi_i'^{-1}(x(r))\rvert \prod_k T_i^{-1}(k)$

any monotonic, differentiable functions

1D function. Parametrize with spline functions

From Gaussian Random Fields, Lognormal Fields to



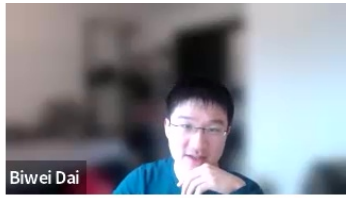
	NF transformation	Probability distribution function
Gaussian Random Fields	$\delta_{\rm GRF}(k) = \sqrt{P(k)}\delta_z(k)$	$p(\delta_{\rm GRF}) = \frac{1}{\sqrt{(2\pi)^n \prod_i P(k_i)}} \exp(-\sum_i \frac{\lvert\delta_{\rm GRF}\rvert^2}{2P(k_i)})$
Lognormal Fields	$\delta_{\rm LN} = \exp(\delta_{\rm GRF}) - C$	$p(\delta_{\rm LN}) = p(\delta_{\rm GRF}) \lvert\frac{\partial \delta_{\rm GRF}}{\partial \delta_{\rm LN}}\rvert$
Generalization	$f_i(x) = \Psi_i(\mathcal{F}^{-1} T_i(k) \mathcal{F} x)$	$p(f_i(x)) = p(x) \prod_r \lvert\Psi_i'^{-1}(x(r))\rvert \prod_k T_i^{-1}(k)$

any monotonic, differentiable functions

1D function. Parametrize with spline functions

Translation and rotation symmetry
CNN-like architecture

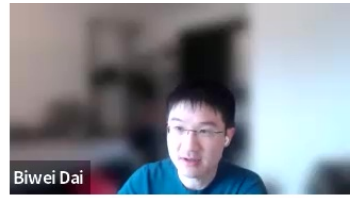
From Gaussian Random Fields, Lognormal Fields to



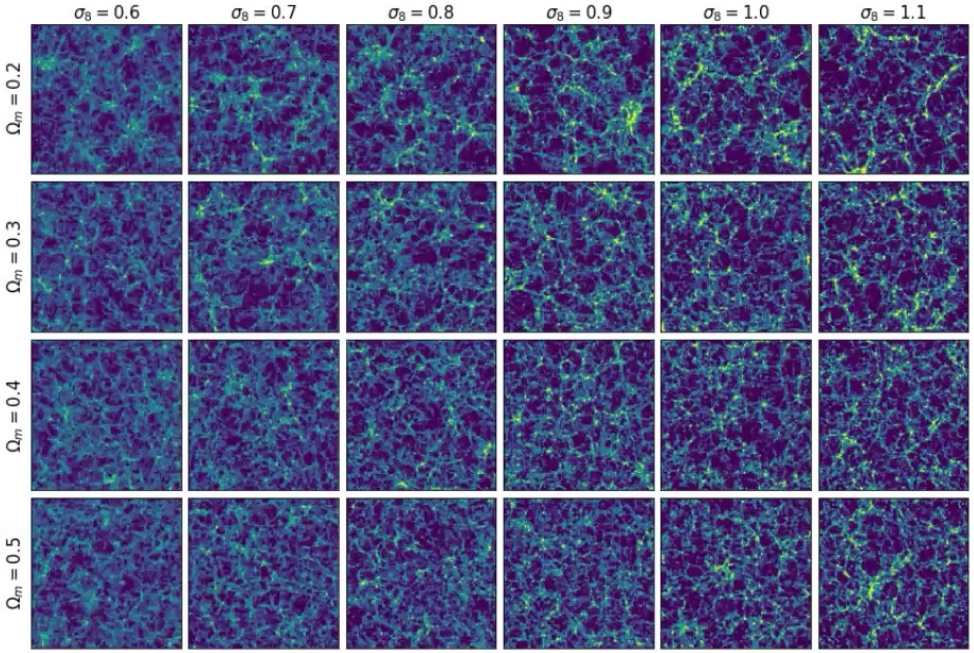
	NF transformation	Probability distribution function
Gaussian Random Fields	$\delta_{\rm GRF}(k) = \sqrt{P(k)}\delta_z(k)$	$p(\delta_{\rm GRF}) = \frac{1}{\sqrt{(2\pi)^n \prod_i P(k_i)}} \exp(-\sum_i \frac{\lvert\delta_{\rm GRF}\rvert^2}{2P(k_i)})$
Lognormal Fields	$\delta_{\rm LN} = \exp(\delta_{\rm GRF}) - C$	$p(\delta_{\rm LN}) = p(\delta_{\rm GRF}) \lvert\frac{\partial \delta_{\rm GRF}}{\partial \delta_{\rm LN}}\rvert$
Generalization	$f_i(x) = \Psi_i(\mathcal{F}^{-1} T_i(k) \mathcal{F} x)$	$p(f_i(x)) = p(x) \prod_r \lvert\Psi_i'^{-1}(x(r))\rvert \prod_k T_i^{-1}(k)$
	any monotonic, differentiable functions	**1D** function. Parametrize with spline functions
		Translation and rotation symmetry CNN-like architecture
Translation and Rotation Equivariant Normalizing Flow (TRENF)	$f = f_1 \circ f_2 \circ \cdots \circ f_n$	$p(f(x)) = p(x) \prod_i \lvert f_i'(x)\rvert^{-1}$

Dai, B. and Seljak, U., 2022. Translation and rotation equivariant normalizing flow (TRENF) for optimal cosmological analysis. *Monthly Notices of the Royal Astronomical Society*, *516*(2), pp.2363-2373.

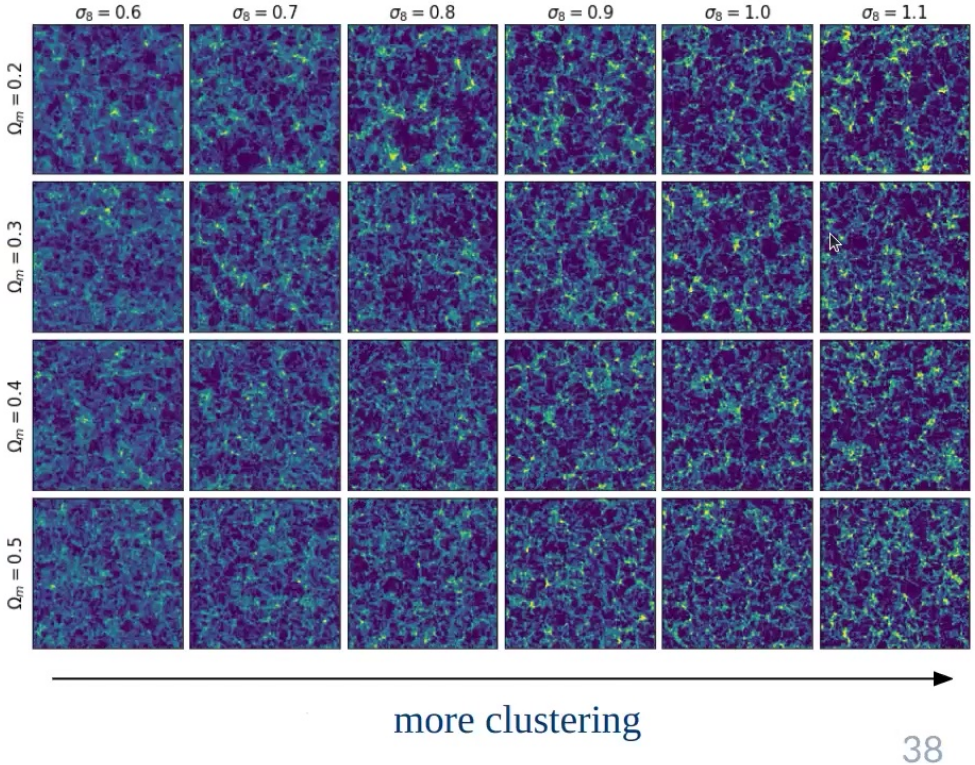
Results -- Samples



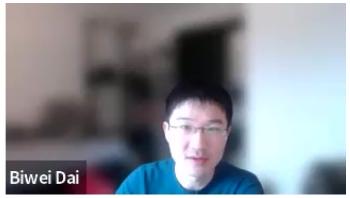
- data:



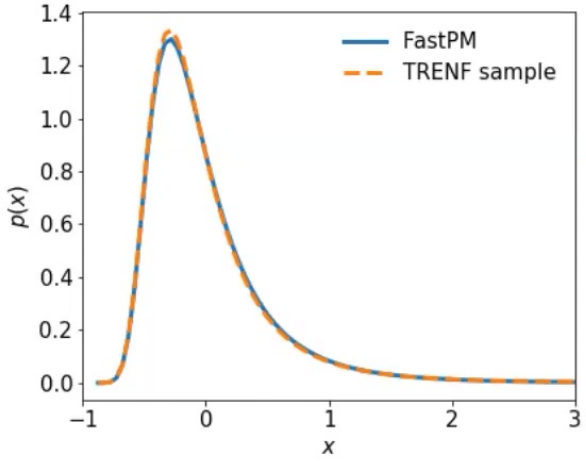
- TRENF samples:



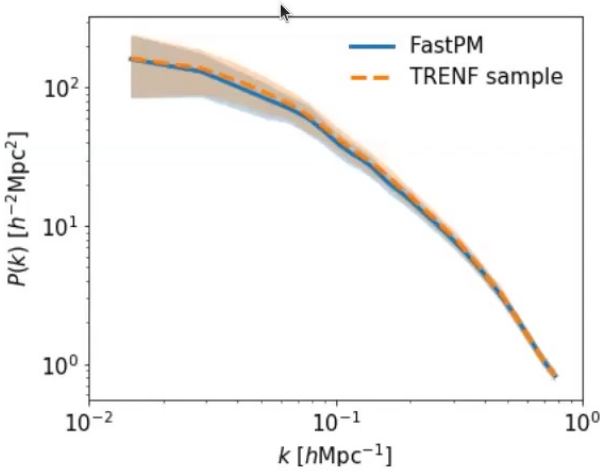
Results -- Summary Statistics of Samples



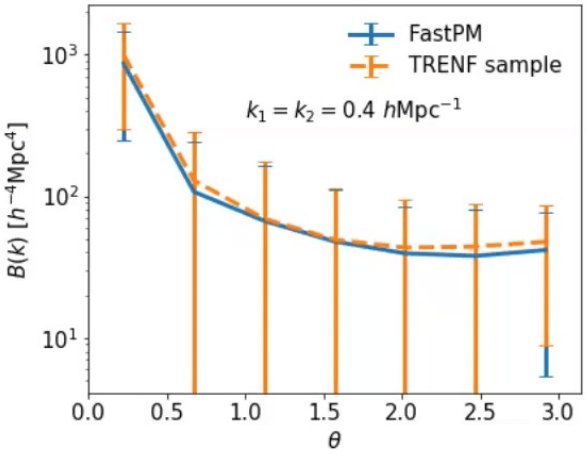
- 1-D probability distribution function



- power spectrum:

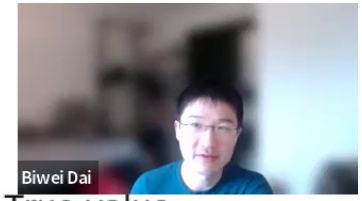


- bispectrum:

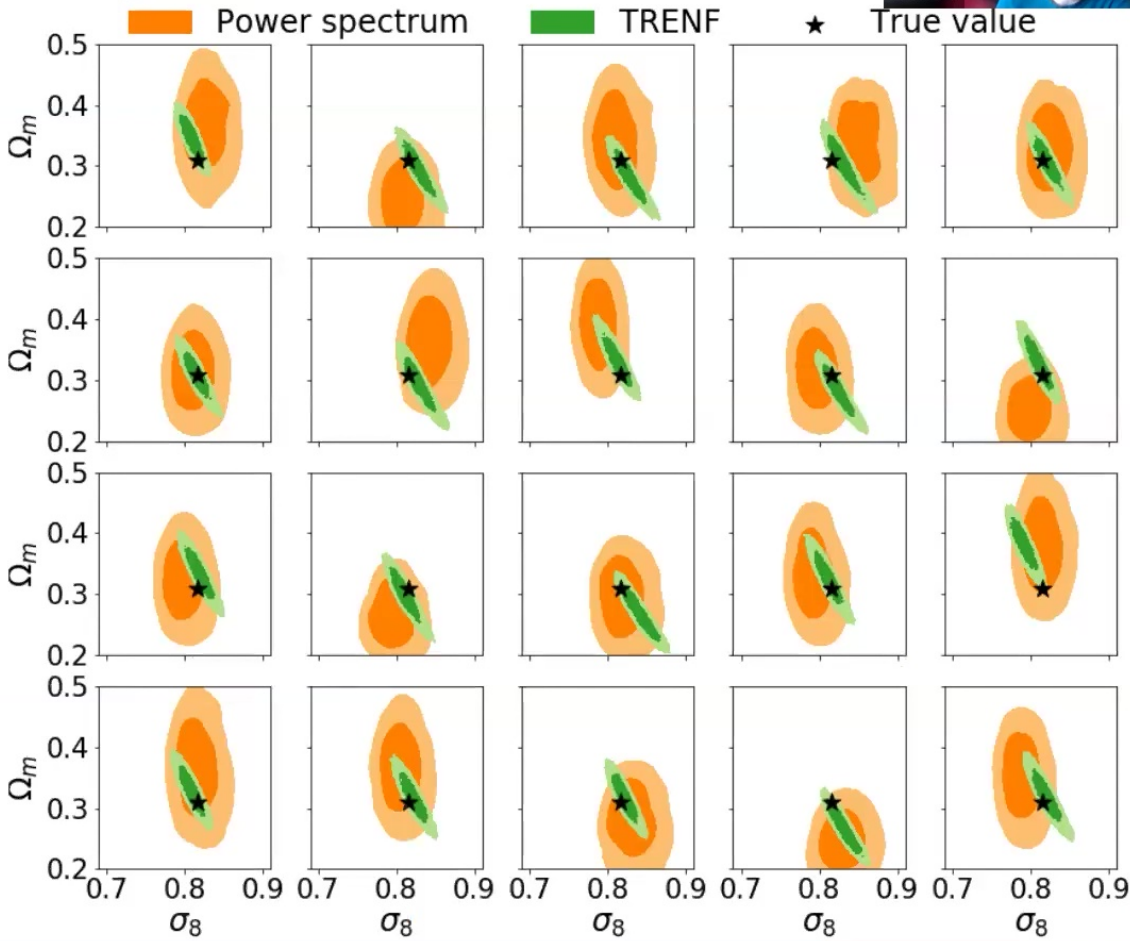


Measured over 10000 samples

Results -- Posterior Analysis

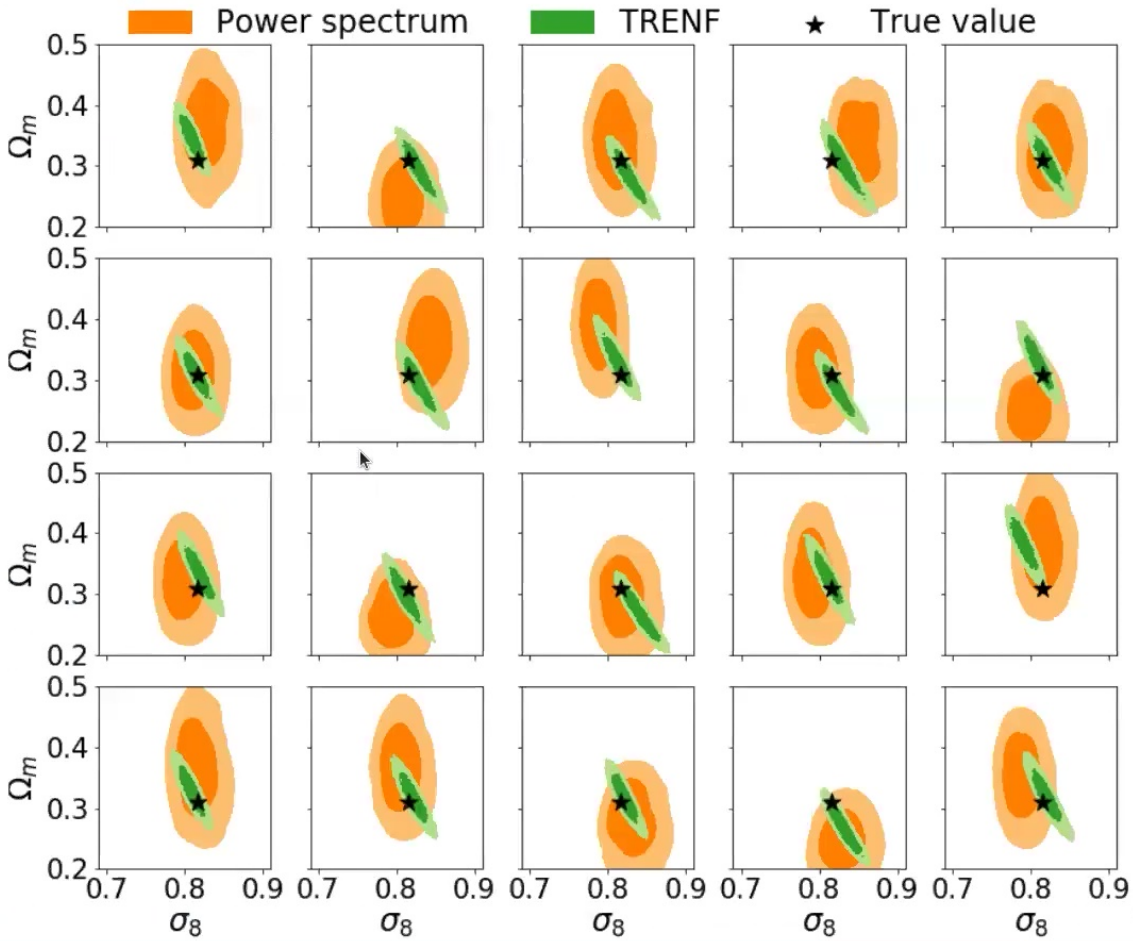


- Full posterior distribution via p(y|x)=p(x|y)p(y)/p(x)
- Other LFI or SBI often use suboptimal summary statistics S and build their p(y|S) or p(S|y)
- Figure of merit (inverse of the area of the 68% confidence region):
  - **power spectrum: ~ 176**
  - **TRENF: ~ 995**
- The posterior is reliable: On 100 test data, 65 cases the true cosmology is within the 68% contour, and 95 cases the true cosmology is within the 95% region.

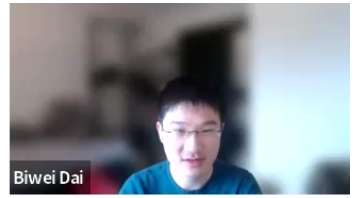


Results -- Posterior Analysis

- Full posterior distribution via p(y|x)=p(x|y)p(y)/p(x)
- Other LFI or SBI often use suboptimal summary statistics S and build their p(y|S) or p(S|y)
- Figure of merit (inverse of the area of the 68% confidence region):
  - **power spectrum: ~ 176**
  - **TRENF: ~ 995**
- The posterior is reliable: On 100 test data, 65 cases the true cosmology is within the 68% contour, and 95 cases the true cosmology is within the 95% region.



Multiscale flow — motivation

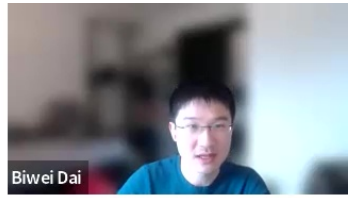


- TRENF works well on Gaussian and mildly non-Gaussian cases, but it's not expressive enough for highly non-Gaussian data

- Multiscale analysis for robust constraints
  - Different scales are governed by different physics / systematics: the numerical / astrophysical effects normally happens on small scales, and PSF may influence very large scales
  - Separate and compare the information (likelihood) of different scales, and identify the part of the data that is contaminated by systematics

Dai, B. and Seljak, U., 2023. Multiscale Flow for Robust and Optimal Cosmological Analysis. *arXiv preprint arXiv:2306.04689*.

(Haar) Wavelet Decomposition



- Recursively apply low-pass filters (scaling functions) and high-pass filters (wavelet functions) to the data. In each iteration, the data x_n with resolution 2^n is decomposed into a low-resolution approximation x_{n-1}, and detail coefficients of the remaining signal $x_{n-1,extra}$

- Haar wavelet: most localized kernel
  to better handle the survey masks

The Haar wavelet's mother wavelet function $\psi(t)$ can be described as

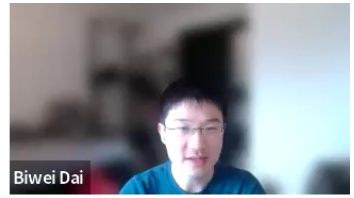
$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2}, \\ -1 & \frac{1}{2} \le t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Its scaling function $\varphi(t)$ can be described as

$$\varphi(t) = \begin{cases} 1 & 0 \le t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

- In 1D, an array $\{a_1, a_2, a_3, a_4, \ldots, a_{2n}\}$ is decomposed into:
  - Low pass: $\{a_1+a_2, a_3+a_4, \ldots, a_{2n-1}+a_{2n}\}$
  - High pass: $\{a_1-a_2, a_3-a_4, \ldots, a_{2n-1}-a_{2n}\}$

(Haar) Wavelet Decomposition



- Recursively apply low-pass filters (scaling functions) and high-pass filters (wavelet functions) to the data. In each iteration, the data x_n with resolution 2^n is decomposed into a low-resolution approximation x_{n-1}, and detail coefficients of the remaining signal $x_{n-1,extra}$

- Haar wavelet: most localized kernel
  to better handle the survey masks

The Haar wavelet's mother wavelet function $\psi(t)$ can be described as

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2}, \\ -1 & \frac{1}{2} \le t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

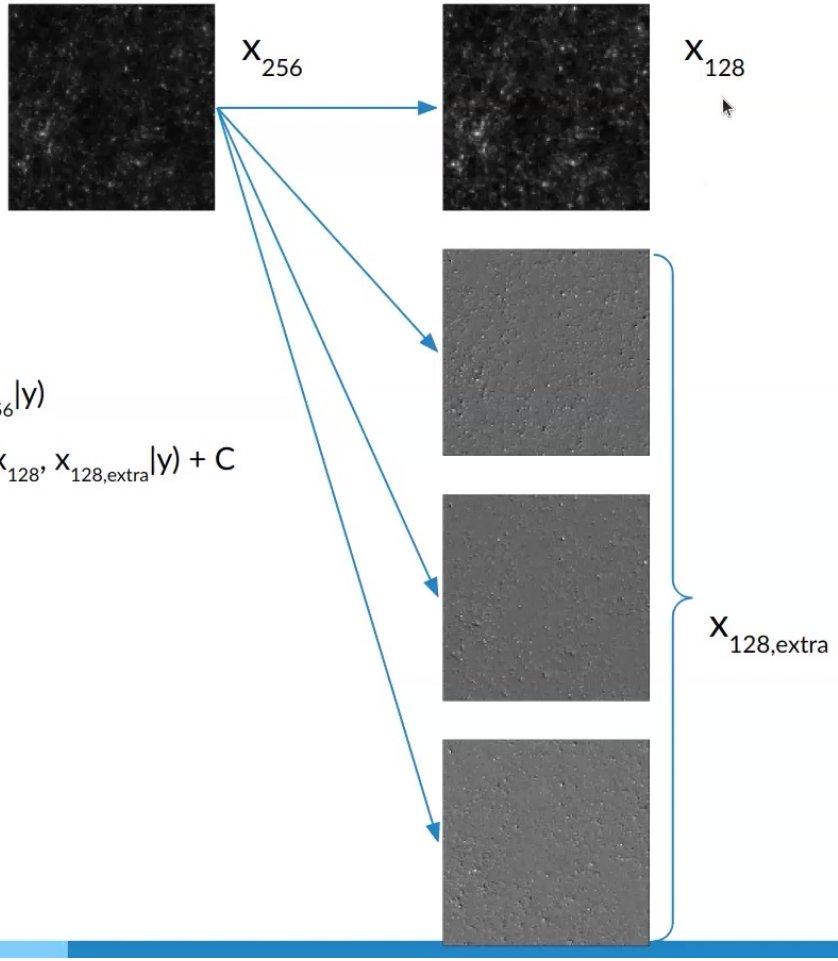
Its scaling function $\varphi(t)$ can be described as

$$\varphi(t) = \begin{cases} 1 & 0 \le t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

- In 1D, an array $\{a_1, a_2, a_3, a_4, \ldots, a_{2n}\}$ is decomposed into:
  - Low pass: $\{a_1+a_2, a_3+a_4, \ldots, a_{2n-1}+a_{2n}\}$
  - High pass: $\{a_1-a_2, a_3-a_4, \ldots, a_{2n-1}-a_{2n}\}$

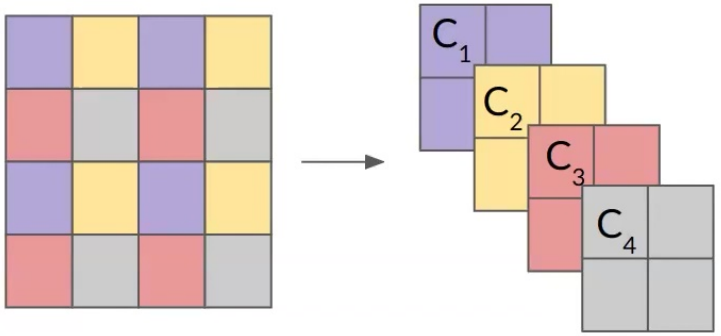
Multiscale flow

- Consider a cosmological field with 256^2 resolution:



$\log p(x_{256}|y)$

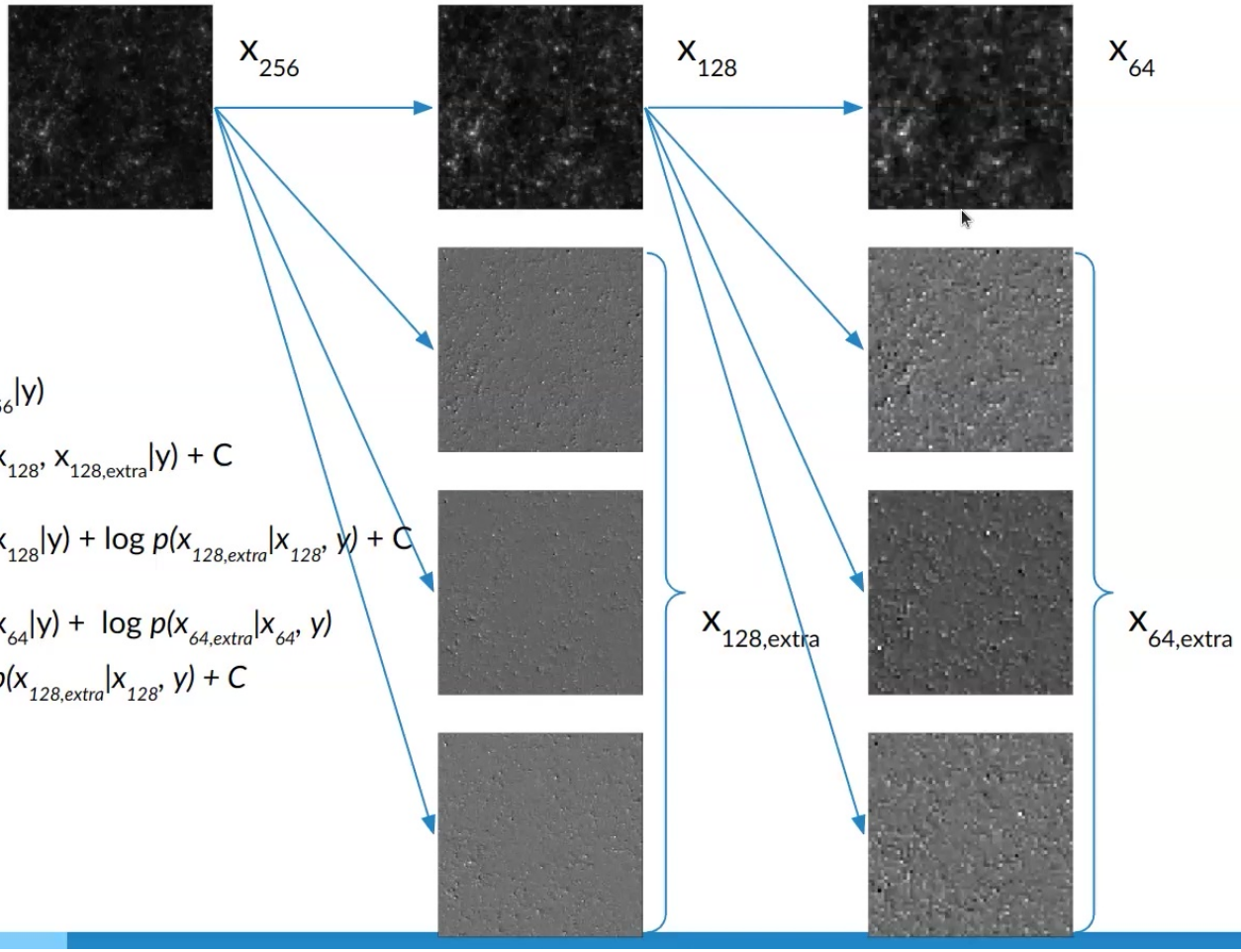
$= \log p(x_{128}, x_{128,extra}|y) + C$



$$\begin{bmatrix} x_{128} \\ x^1_{128,\text{extra}} \\ x^2_{128,\text{extra}} \\ x^3_{128,\text{extra}} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix}$$

Multiscale flow

- Consider a cosmological field with 256^2 resolution:



$\log p(x_{256}|y)$

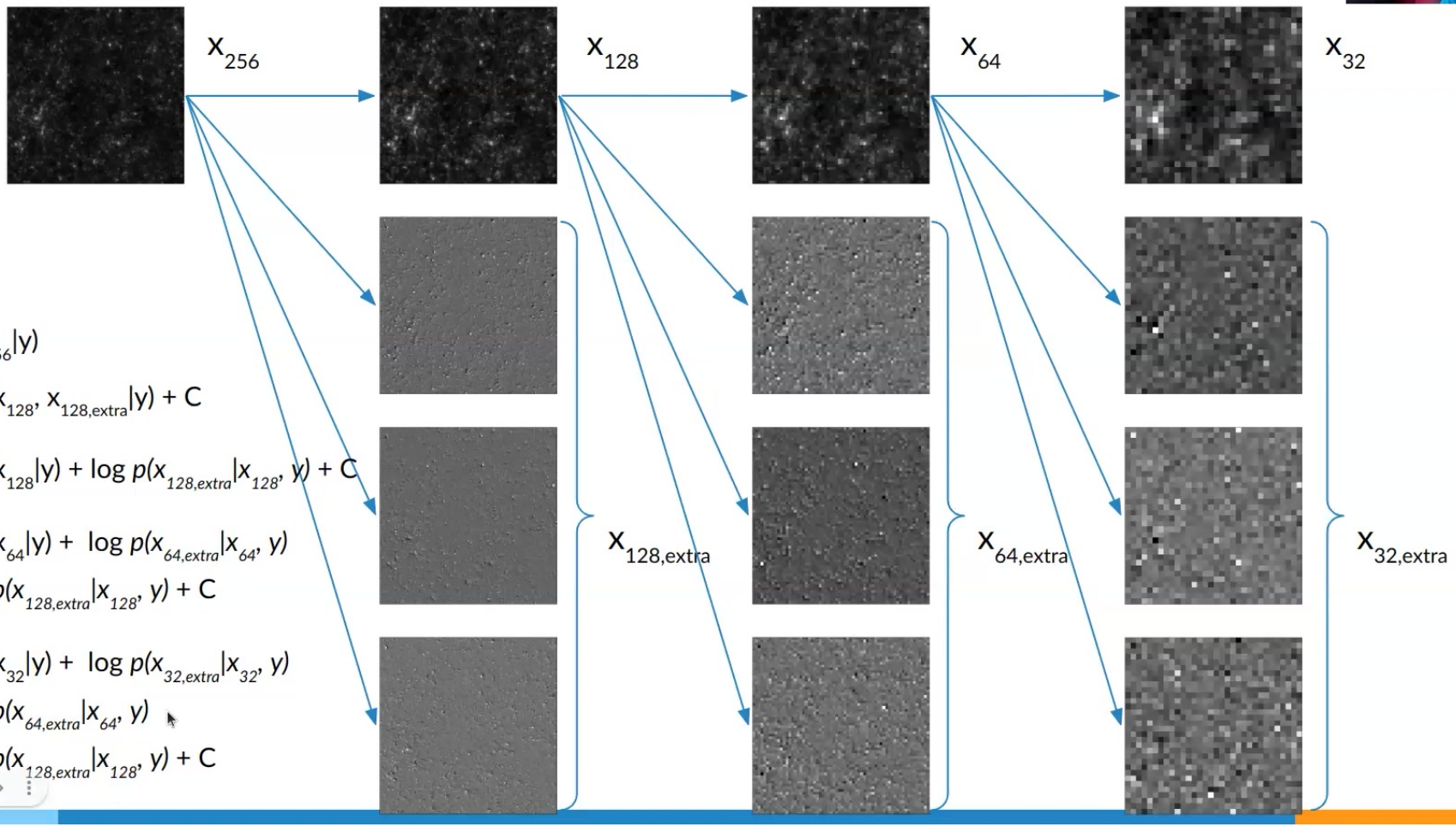
$= \log p(x_{128}, x_{128,extra}|y) + C$

$= \log p(x_{128}|y) + \log p(x_{128,extra}|x_{128}, y) + C$

$= \log p(x_{64}|y) + \log p(x_{64,extra}|x_{64}, y)$
$+ \log p(x_{128,extra}|x_{128}, y) + C$

Multiscale flow

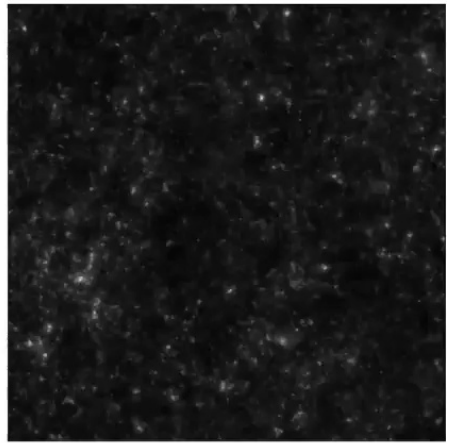
- Consider a cosmological field with 256^2 resolution:



$$\log p(x_{256}|y)$$
$$= \log p(x_{128}, x_{128,extra}|y) + C$$
$$= \log p(x_{128}|y) + \log p(x_{128,extra}|x_{128}, y) + C$$
$$= \log p(x_{64}|y) + \log p(x_{64,extra}|x_{64}, y) + \log p(x_{128,extra}|x_{128}, y) + C$$
$$= \log p(x_{32}|y) + \log p(x_{32,extra}|x_{32}, y) + \log p(x_{64,extra}|x_{64}, y) + \log p(x_{128,extra}|x_{128}, y) + C$$

Experiments

- Dataset: simulated WL convergence maps (Zorrilla Matilla et al. 2016)
  - 512^2 resolution, 3.5 x 3.5 deg^2 field of view
  - Varying Ω_m and σ_8
  - Ray-tracing to redshift 1 in N-body simulations
  - Baryon correction model (4 baryon parameters)

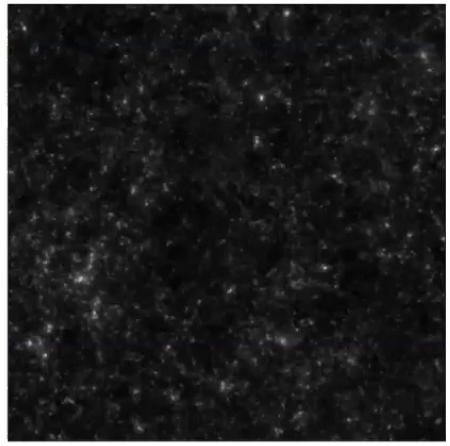


- Decompose the likelihood into 4 terms:

$$\log p(x_{512}|y) = \log p(x_{64}|y) + \log p(x_{64,extra}|x_{64}, y) + \log p(x_{128,extra}|x_{128}, y) + \log p(x_{256,extra}|x_{256}, y)$$

Experiments

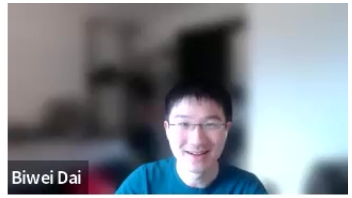
- Dataset: simulated WL convergence maps (Zorrilla Matilla et al. 2016)
  - 512^2 resolution, 3.5 x 3.5 deg^2 field of view
  - Varying Ω_m and σ_8
  - Ray-tracing to redshift 1 in N-body simulations
  - Baryon correction model (4 baryon parameters)



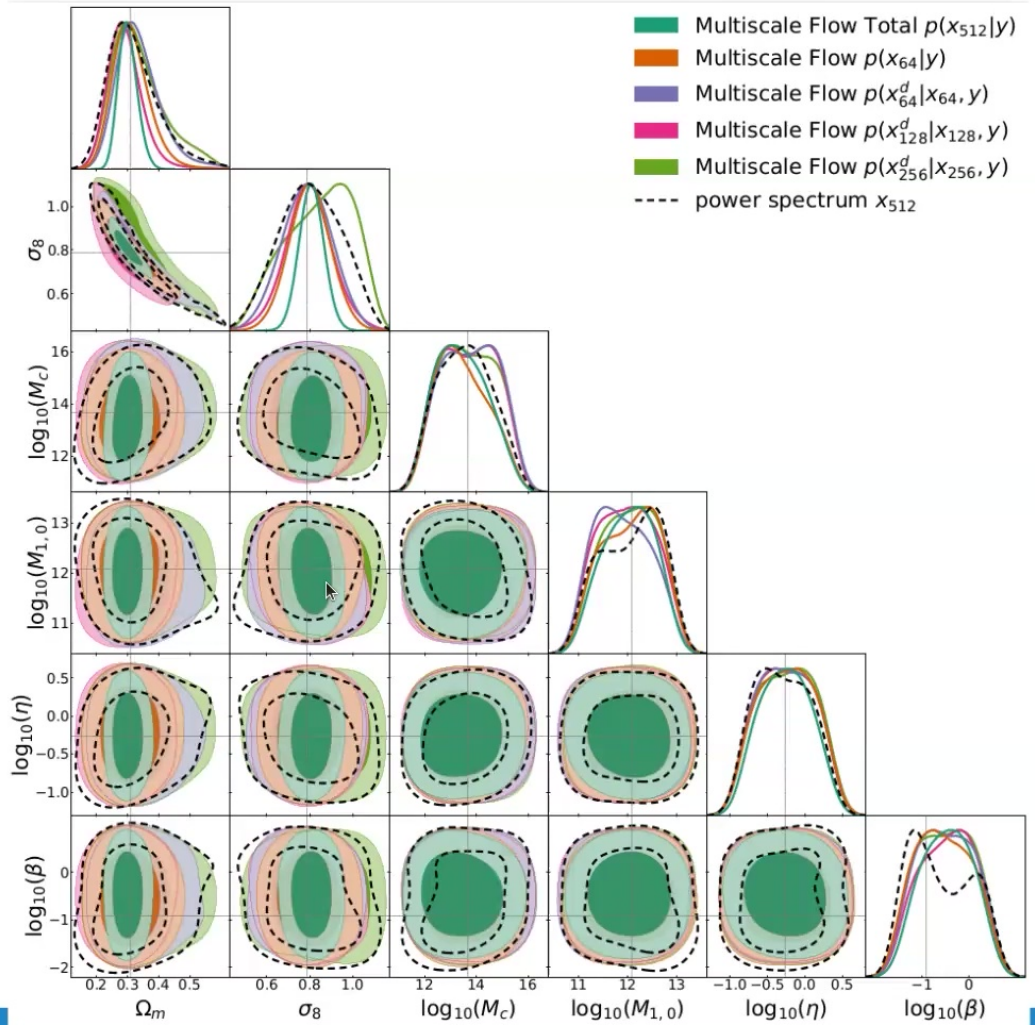
- Decompose the likelihood into 4 terms:

$$\log p(x_{512}|y) = \log p(x_{64}|y) + \log p(x_{64,extra}|x_{64}, y) + \log p(x_{128,extra}|x_{128}, y) + \log p(x_{256,extra}|x_{256}, y)$$

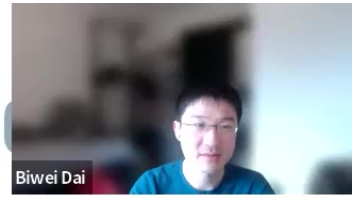
Constraining power on 3.5 x 3.5 deg^2 convergence



- ng = 20 arcmin^{-2}



Constraining power on 3.5 x 3.5 deg^2 convergenc



- Figure of merit: the reciprocal of the 1σ confidence area on the (Ω_m, σ_8) plane

	Method	$n_g = 10\text{arcmin}^{-2}$	$n_g = 20\text{arcmin}^{-2}$	$n_g = 50\text{arcmin}^{-2}$	$n_g = 100\text{arcmin}^{-2}$
Fix baryon parameters at fiducial values	Multiscale Flow $p(x_{512}\|y)$	**166**	**310**	**617**	**1072**
	Multiscale Flow $p(x_{256}\|y)$	164	297	558	947
	Multiscale Flow $p(x_{128}\|y)$	124	214	415	704
	Multiscale Flow $p(x_{64}\|y)$	81	136	247	387
	power spectrum	41(41)	61 (58)	95(87)	127(111)
	CNN[1]	-	(93)	(146)	(194)
Marginalize over baryon parameters	Multiscale Flow $p(x_{512}\|y)$	**149**	**220**	**362**	**521**
	Multiscale Flow $p(x_{256}\|y)$	147	213	341	494
	Multiscale Flow $p(x_{128}\|y)$	112	166	269	398
	Multiscale Flow $p(x_{64}\|y)$	75	113	183	259
	power spectrum	34(33)	48(48)	68(65)	84 (78)
	CNN[2]	-	($\lesssim$ 77)	($\lesssim$ 109)	($\lesssim$ 136)

The numbers in parenthesis are estimated using maps with 1arcmin Gaussian smoothing.

Multiscale Flow posterior is reliable

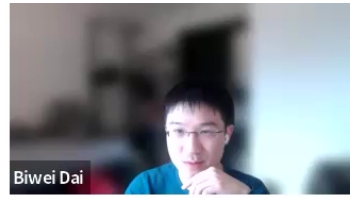
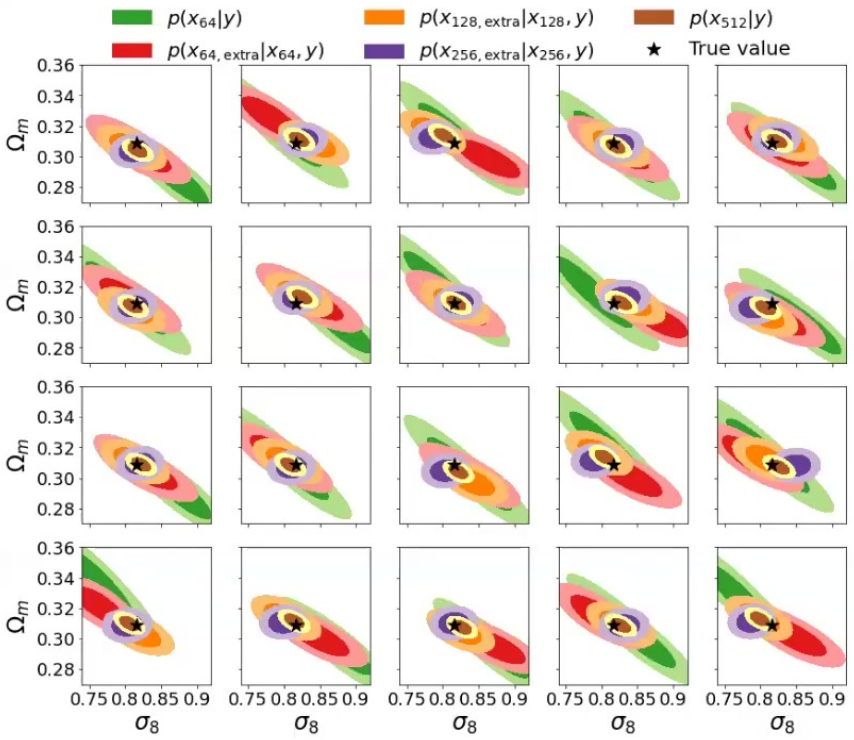


Table 4. Empirical coverage probability of posteriors from different methods, after marginalizing over baryon parameters. We report the percentage of test data that falls within 68% confidence region and 95% confidence regions. A perfectly calibrated posterior should have 68% and 95% test data that falls in these two regions, respectively.

Method	$n_g = 10\text{arcmin}^{-2}$	$n_g = 20\text{arcmin}^{-2}$	$n_g = 50\text{arcmin}^{-2}$	$n_g = 100\text{arcmin}^{-2}$
Multiscale Flow $p(x_{512}\|y)$	72.8%, 96.8%	74.4%, 95.2%	73.6%, 97.6%	66.4%, 97.6%
Multiscale Flow $p(x_{256}\|y)$	70.4%, 96.0%	76.8%, 95.2%	74.4%, 97.6%	68.8%, 96.8%
Multiscale Flow $p(x_{128}\|y)$	76.0%, 96.0%	74.4%, 97.6%	76.0%, 97.6%	73.6%, 96.8%
Multiscale Flow $p(x_{64}\|y)$	80.8%, 95.2%	70.4%, 94.4%	72.0%, 95.2%	74.4%, 95.2%

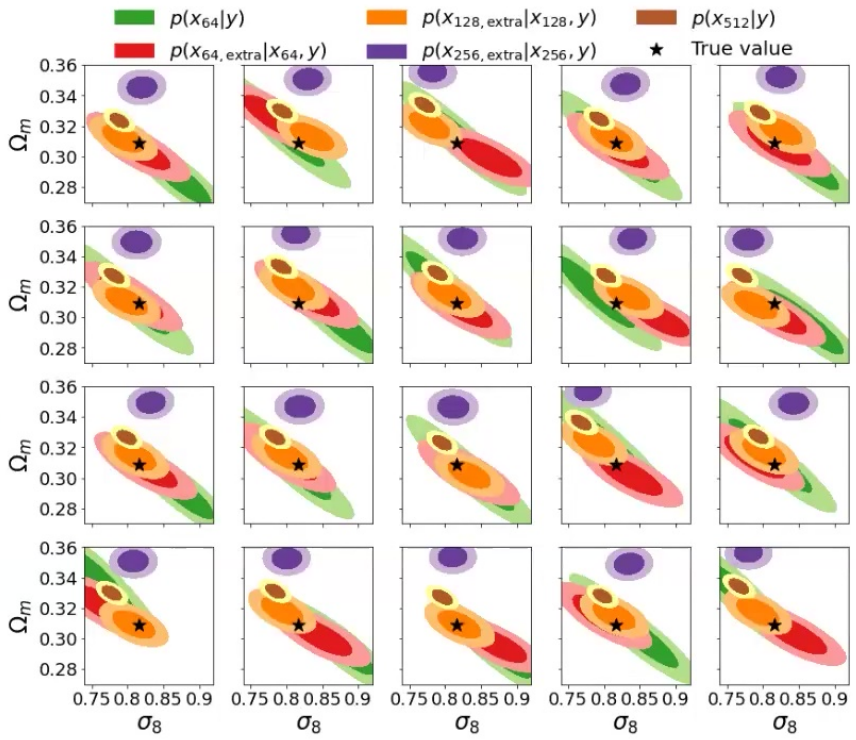
Distribution shift detection — noise miscalibr

- Consistent posteriors from different scales



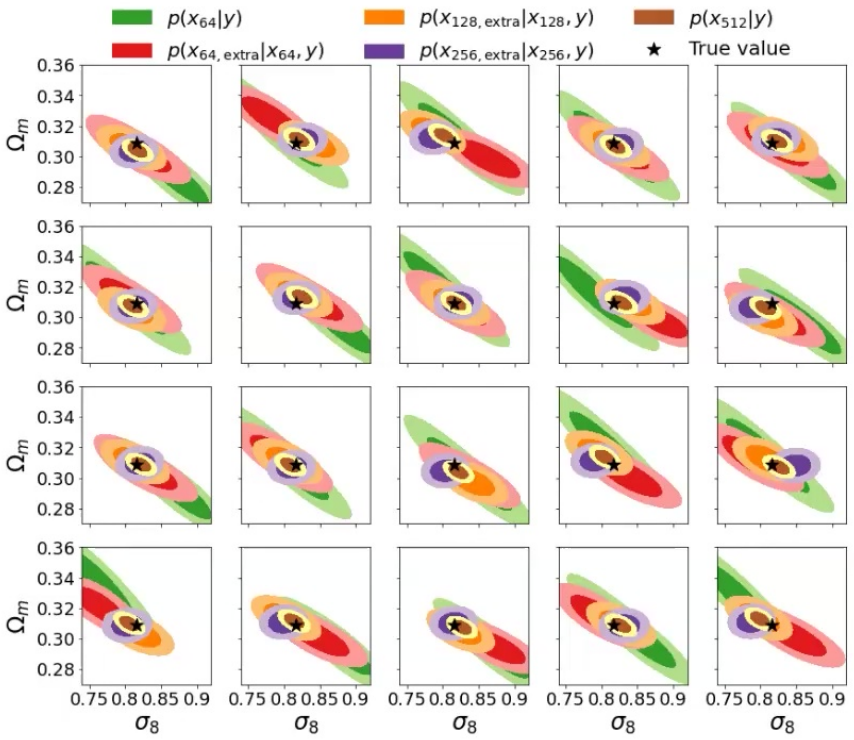
noise miscalibration

- Inconsistent small scale posterior



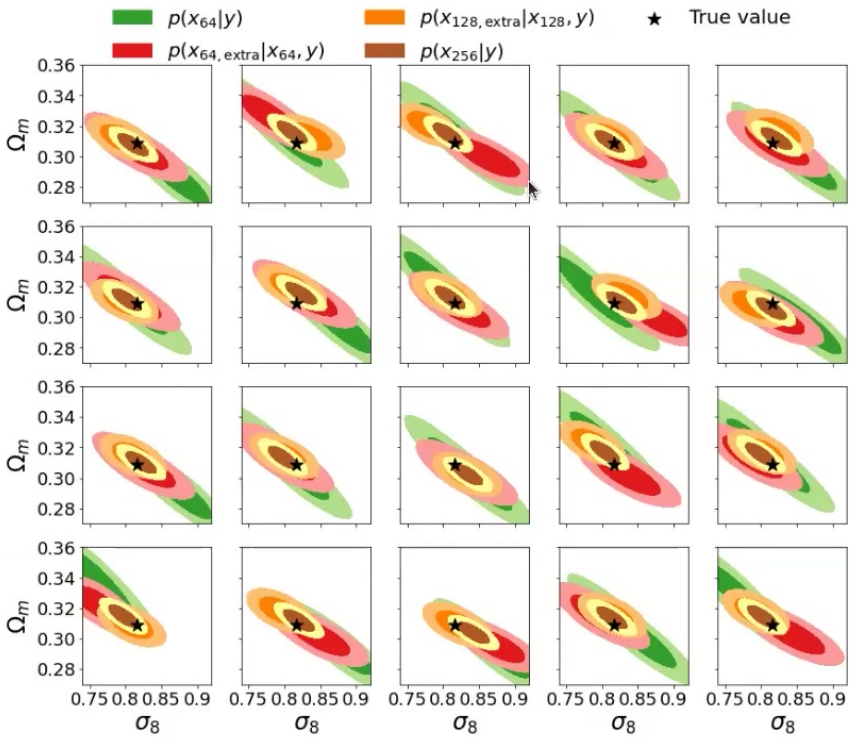
Distribution shift detection — noise miscalibr

- Consistent posteriors from different scales



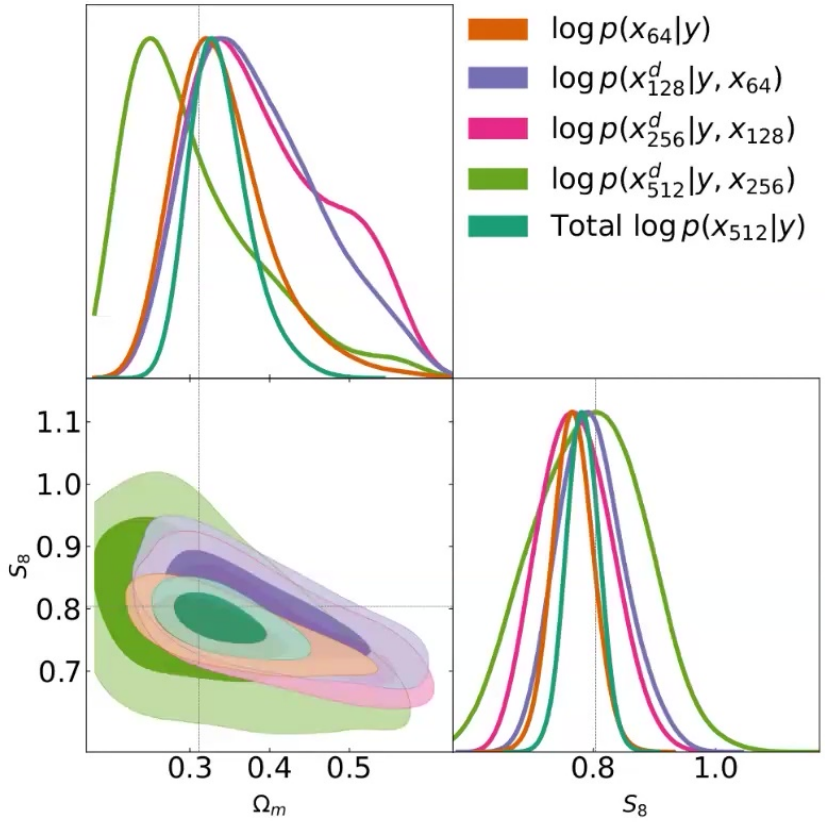
noise miscalibration

- Remove small scale information

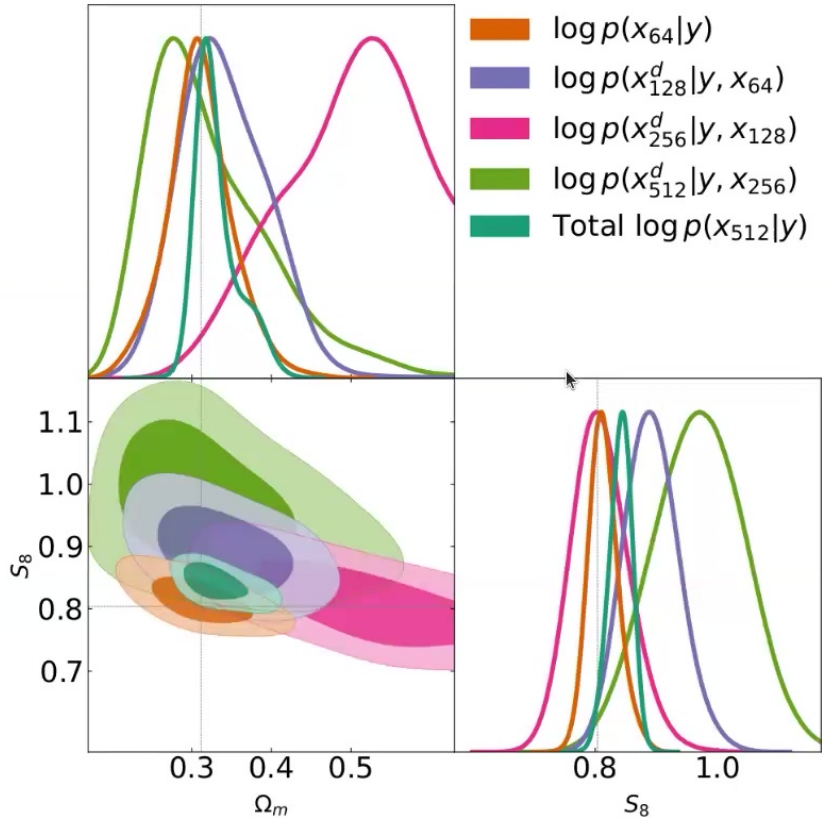


Distribution shift detection — baryon physics

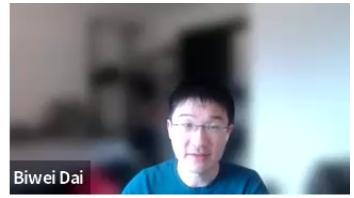
- MF trained on baryon models



- MF trained on DMO simulations

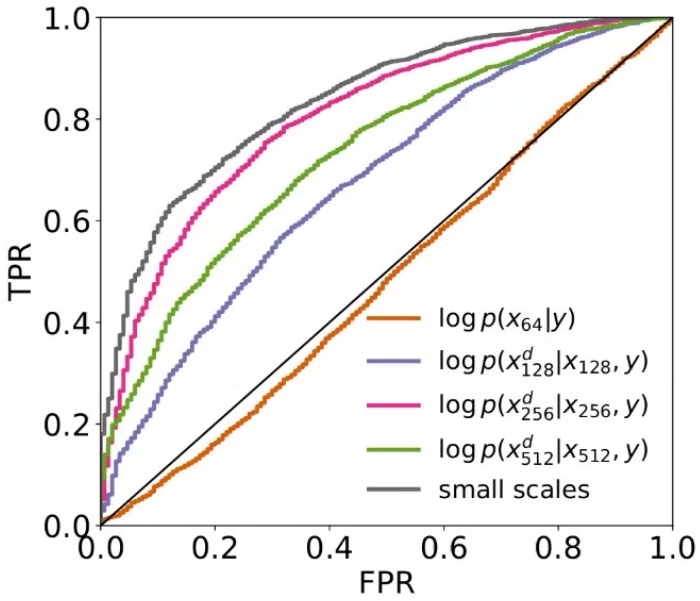


Distribution shift detection — baryon physics



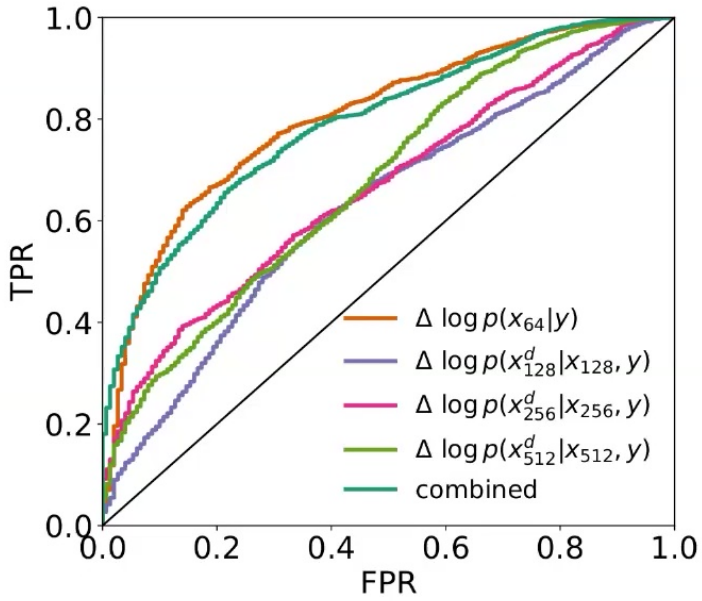
We train Multiscale Flow with DMO maps, and apply it to maps with baryonic physics to test if we can identify this systematic effect (distribution shift) with goodness-of-fit test (left) and consistency test (right)

- ROC curve with logp
(if it's consistent with training simulations)

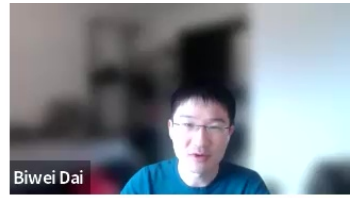


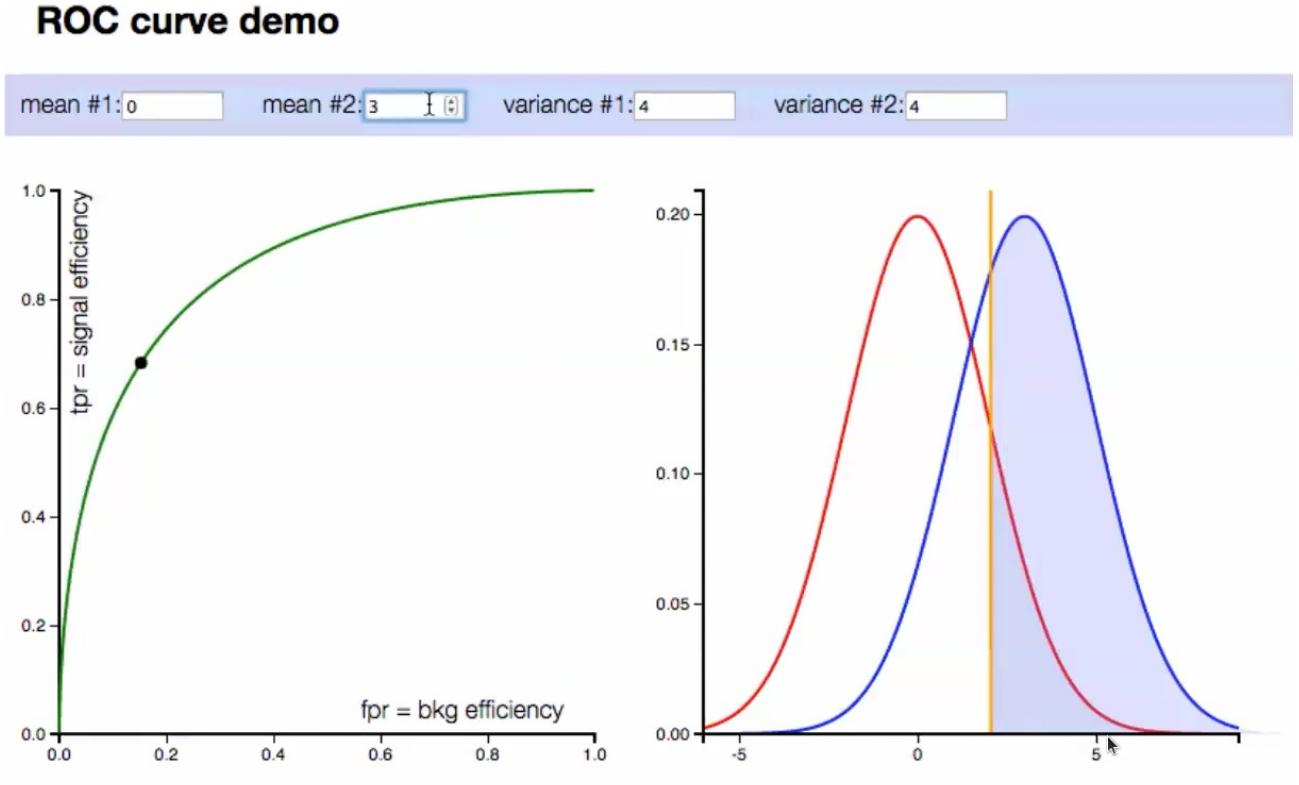
- ROC curve with Δlogp
(if different scales are consistent with each other)

$$\Delta \log p(x_m|y) = \log p(x_m|y_{\mathrm{MAP}}) - \log p(x_m|y_{\mathrm{MAP},m})$$



ROC curve



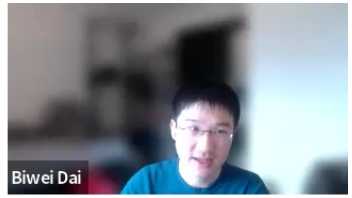


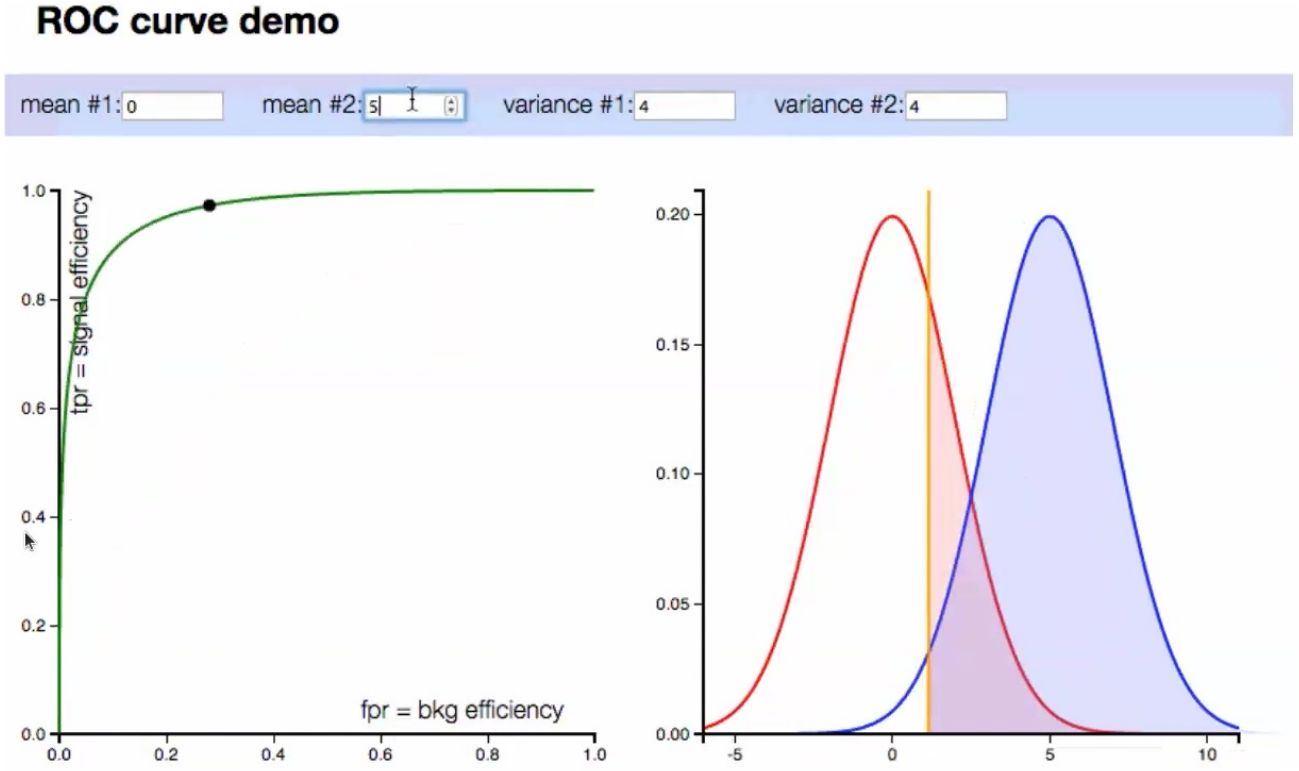
Diagonal line: two distributions completely overlap

The upper left, the better

Credit: https://arogozhnikov.github.io/2015/10/05/roc-curve.html

ROC curve



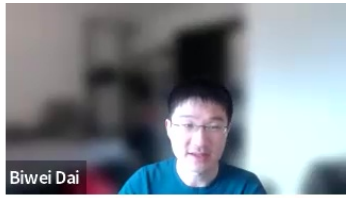


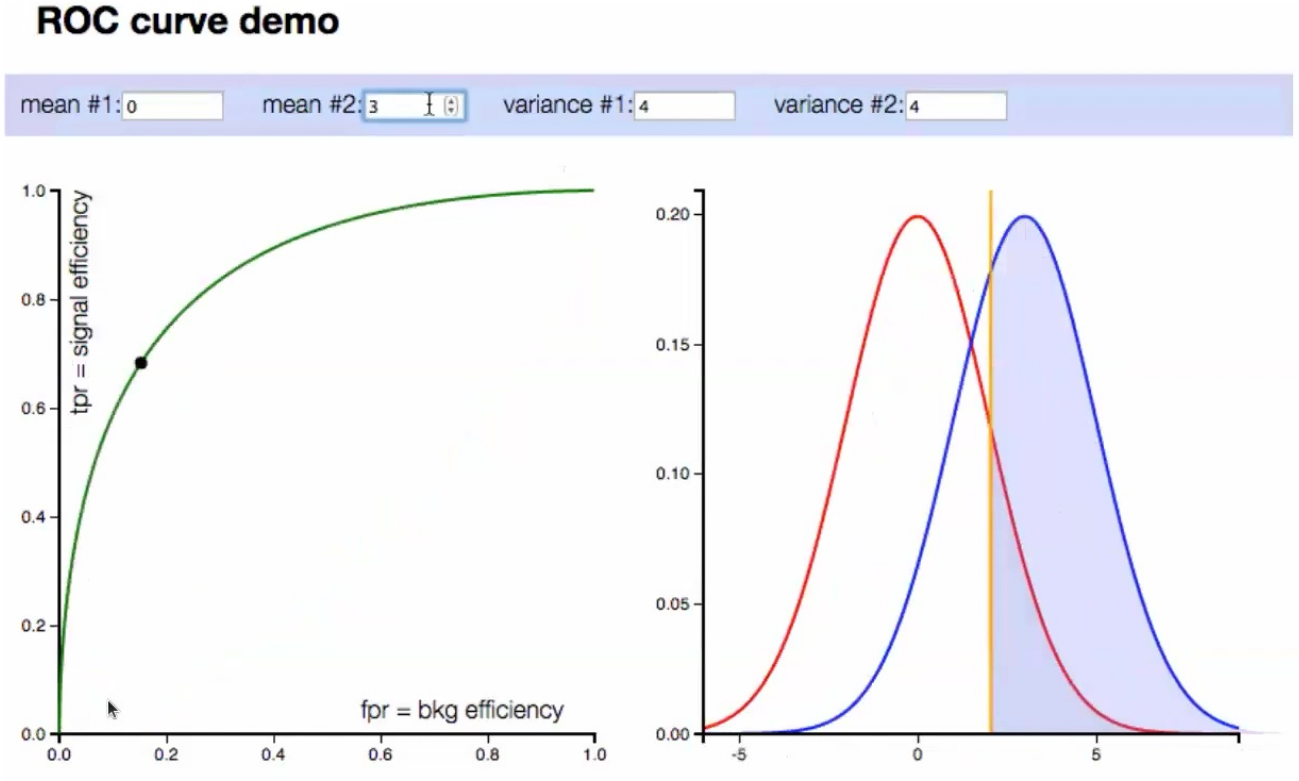
Diagonal line: two distributions completely overlap

The upper left, the better

Credit: https://arogozhnikov.github.io/2015/10/05/roc-curve.html

ROC curve





Diagonal line: two distributions completely overlap

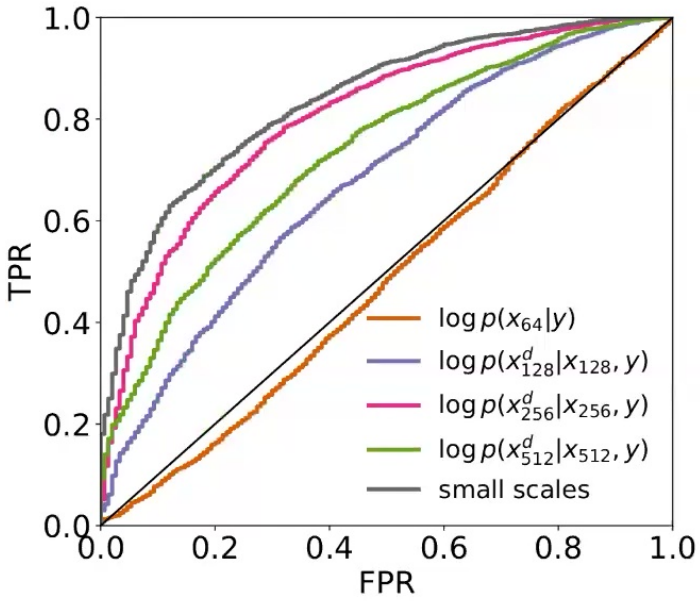
The upper left, the better

Credit: https://arogozhnikov.github.io/2015/10/05/roc-curve.html

Distribution shift detection — baryon physics

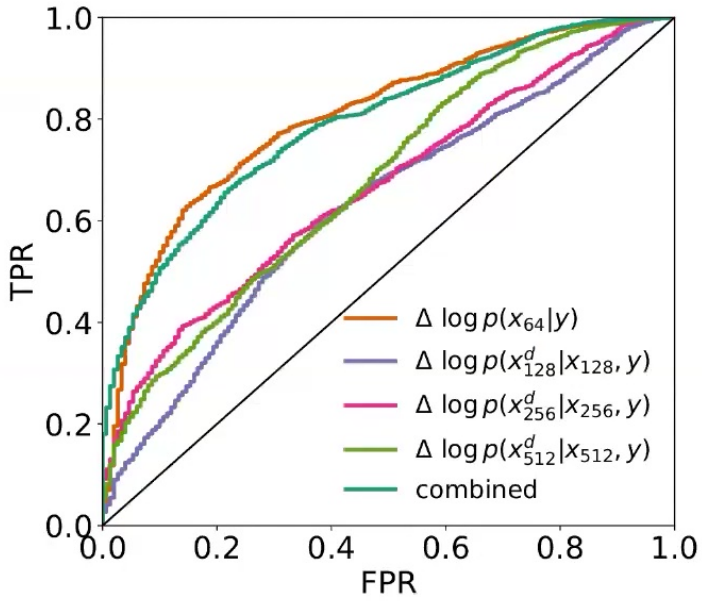
We train Multiscale Flow with DMO maps, and apply it to maps with baryonic physics to test if we can identify this systematic effect (distribution shift) with goodness-of-fit test (left) and consistency test (right)

- ROC curve with logp
(if it's consistent with training simulations)

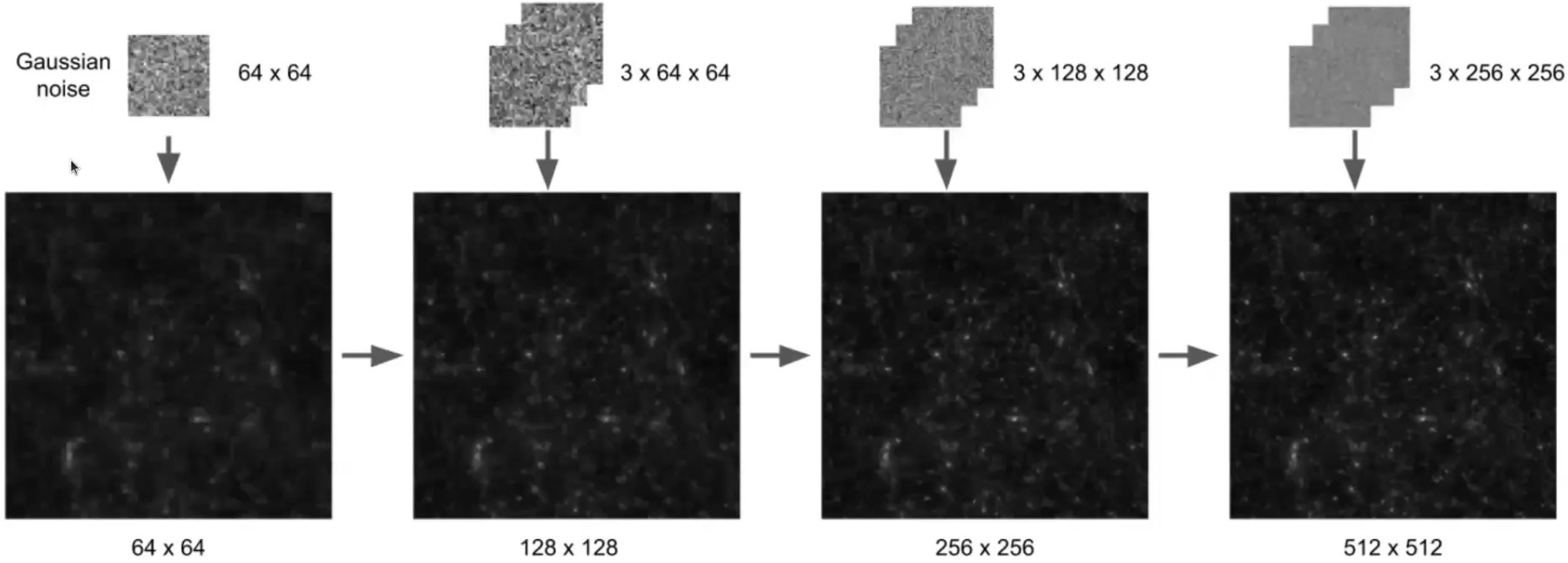


- ROC curve with Δlogp
(if different scales are consistent with each other)

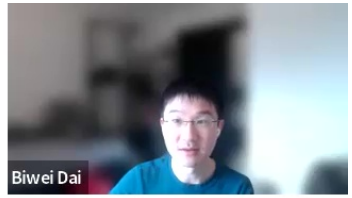
$$\Delta \log p(x_m|y) = \log p(x_m|y_{\mathrm{MAP}}) - \log p(x_m|y_{\mathrm{MAP},m})$$



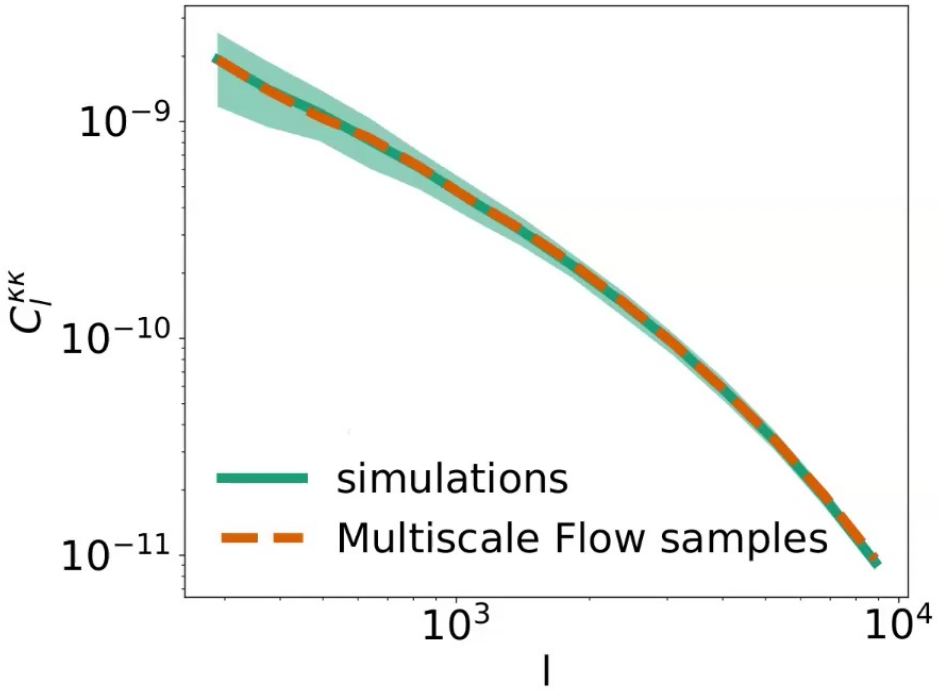
Sample generation & super-resolution



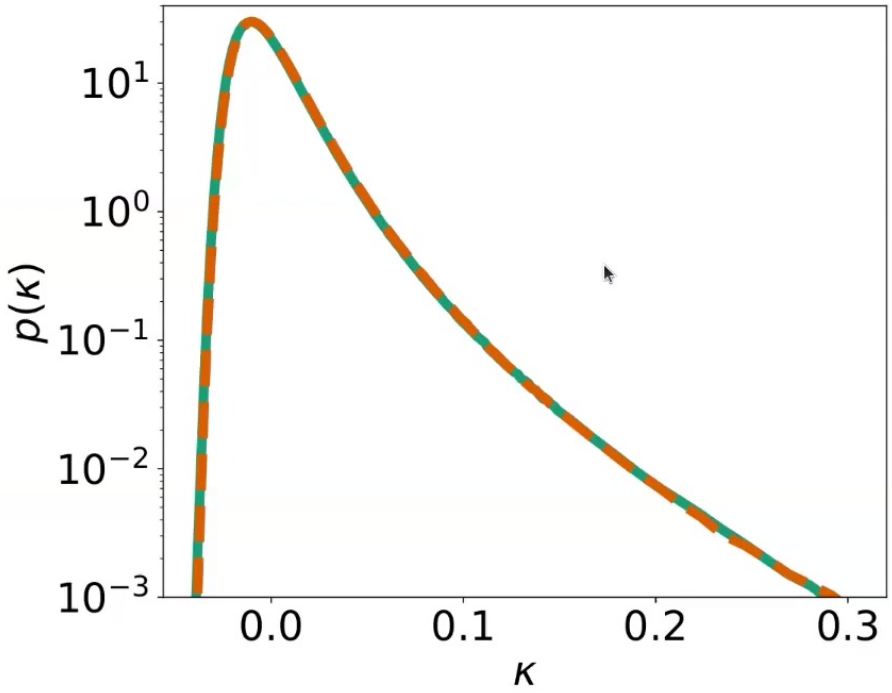
Sample generation & super-resolution



- power spectrum



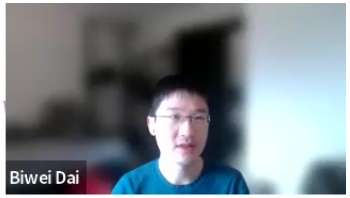
- kappa probability distribution



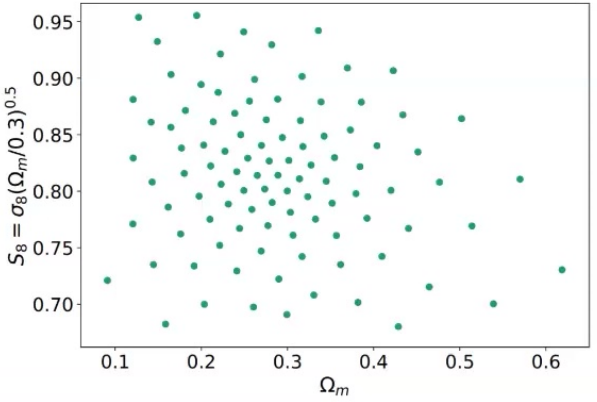
60

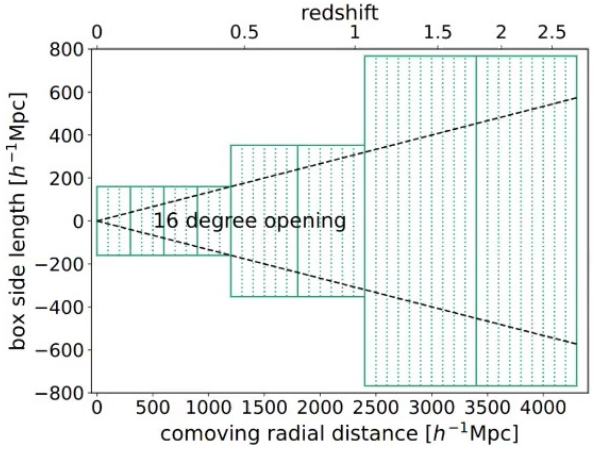
HSC cosmic shear analysis with Multiscale Flo

With *Xiangchong Li, Uroš Seljak and Rachel Mandelbaum*



- Training maps:
  - 100 cosmologies with different Ω_m, σ_8
  - 3 simulation boxes with different sizes and resolutions for different redshifts
  - Ray-tracing with multi-lens-plane algorithm to simulate the cosmic shear at galaxy positions

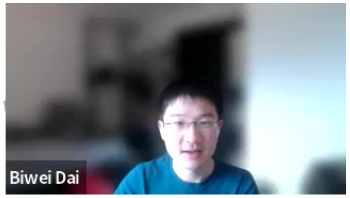




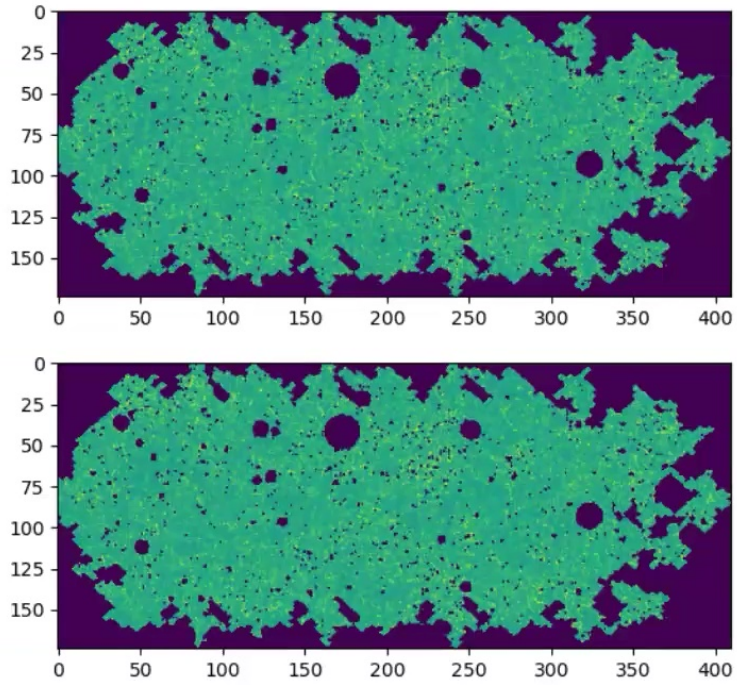
redshift	box size	$N_{\rm particle}$	force resolution	$N_{\rm step}$	simulation code
$0 < z < 0.45$	320 h^{-1}Mpc	960^3	0.03 h^{-1}Mpc	adaptive	MP-Gadget [49]
$0.45 < z < 1.05$	704 h^{-1}Mpc	2816^3	0.125 h^{-1}Mpc	60	FastPM [50]
$1.05 < z < 2.72$	1536 h^{-1}Mpc	1536^3	0.5 h^{-1}Mpc	15	FastPM [50]

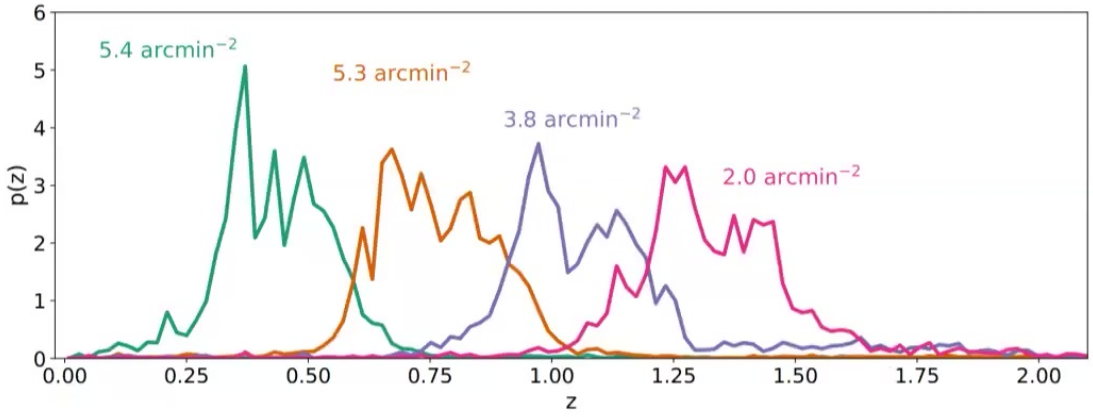
HSC cosmic shear analysis with Multiscale Flo

With *Xiangchong Li, Uroš Seljak and Rachel Mandelbaum*



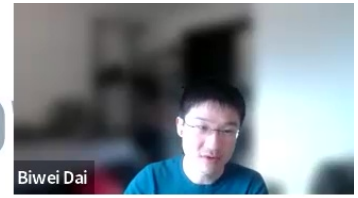
- Training maps:
  - 100 cosmologies with different Ω_m, σ_8
  - 3 simulation boxes with different sizes and resolutions for different redshifts
  - Ray-tracing with multi-lens-plane algorithm to simulate the cosmic shear at galaxy positions





HSC cosmic shear analysis with Multiscale Flo

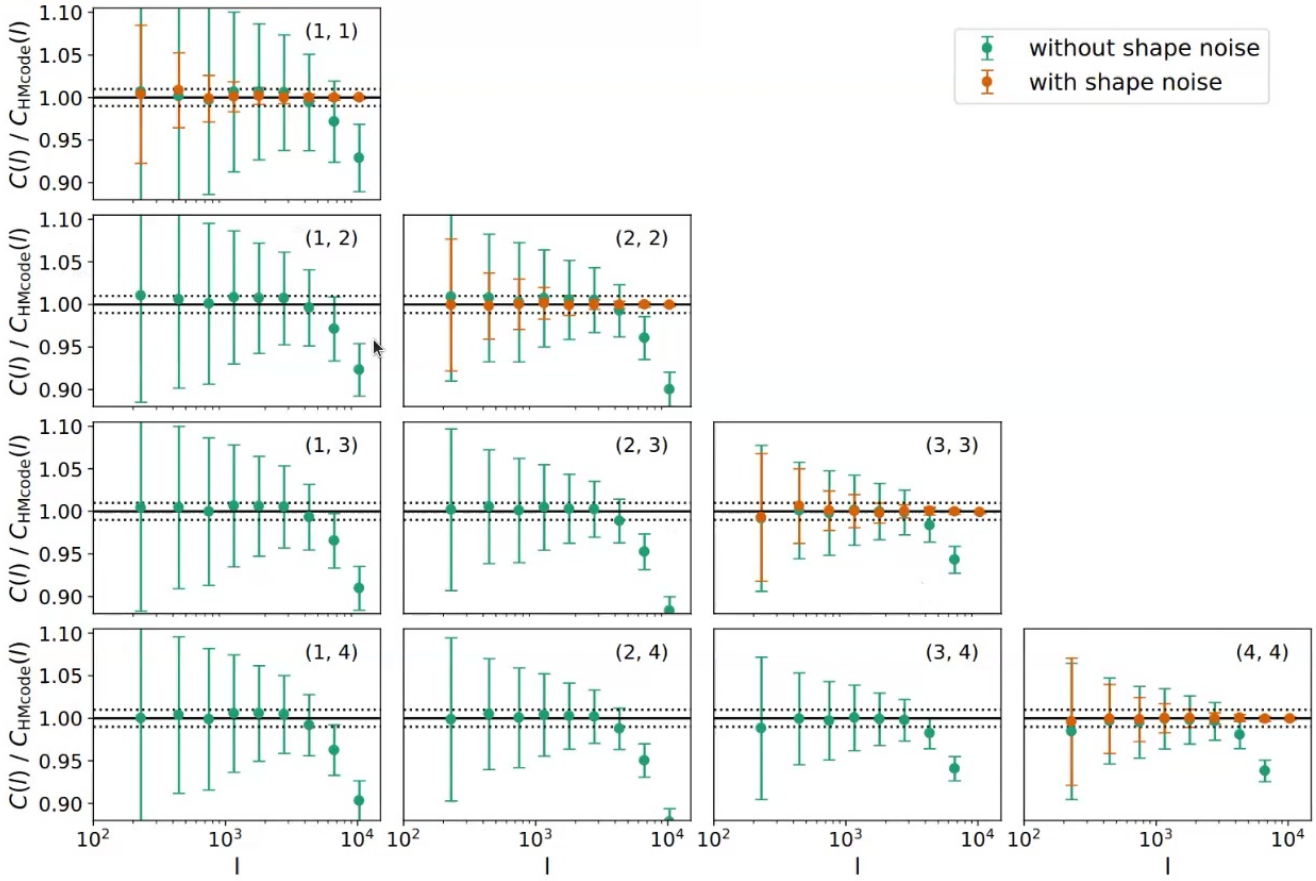
With *Xiangchong Li, Uroš Seljak and Rachel Mandelbaum*



- Training maps:

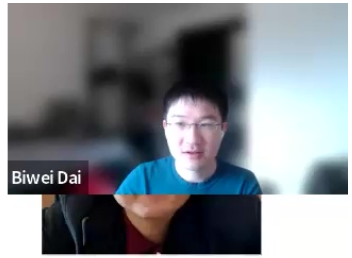
- Consistent with HMcode 2020 within 1% up to l=4000.

- At l>4000, the maps lose power because of FastPM at z>0.45. These scales are dominated by noise.



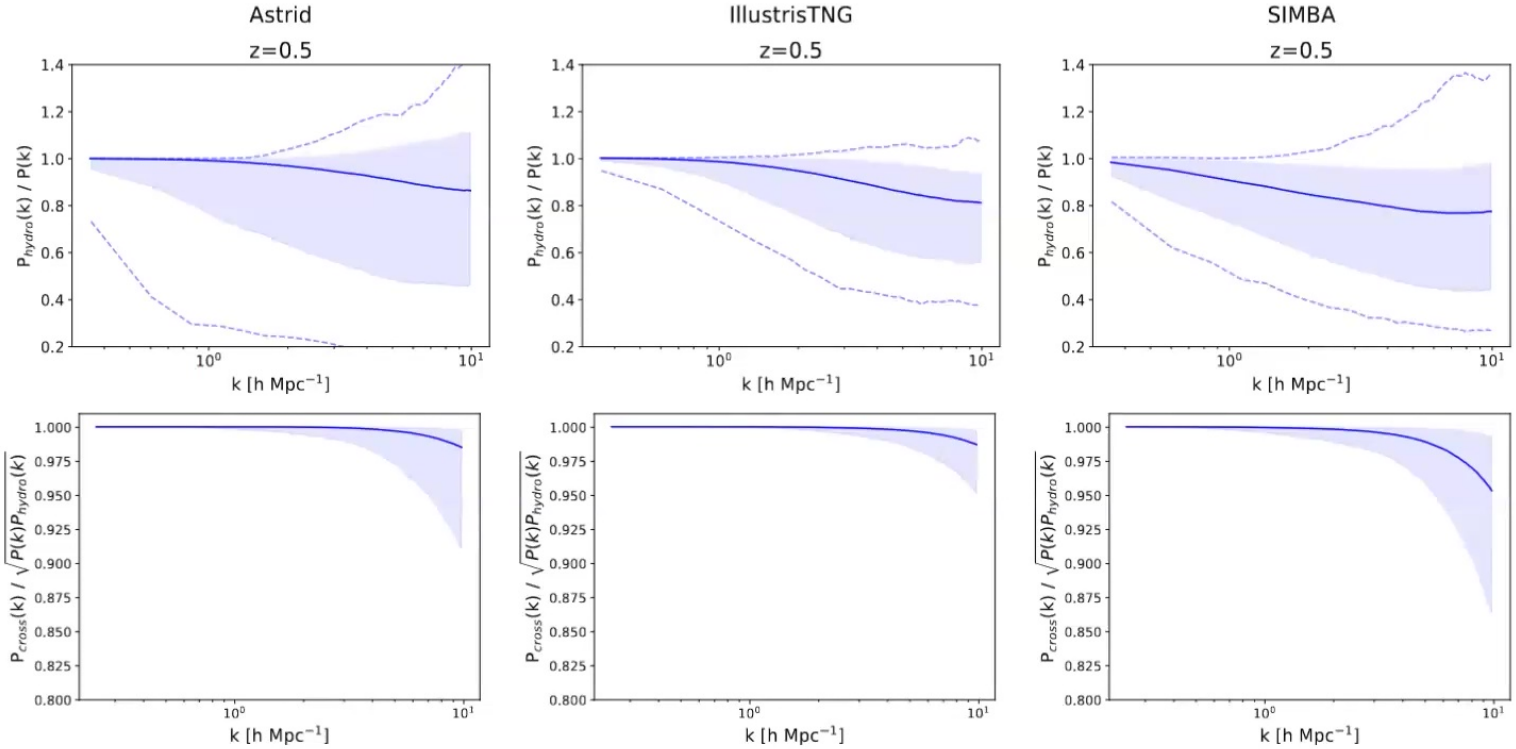
Field-level baryonic effect emulator

With *Francisco Villaescusa-Navarro and Uroš Seljak*



Divij Sharma

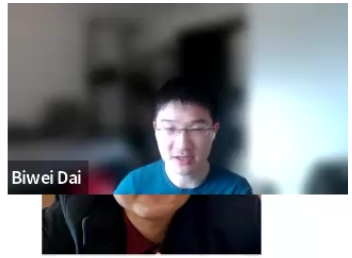
Baryonic effects from CAMELS Simulations (FVN et al. 2022)



- The baryonic effect is mostly on the amplitude of the fourier modes (power spectrum, top panel), not the phase of the fourier modes (cross correlation coefficient, bottom panel)

Field-level baryonic effect emulator

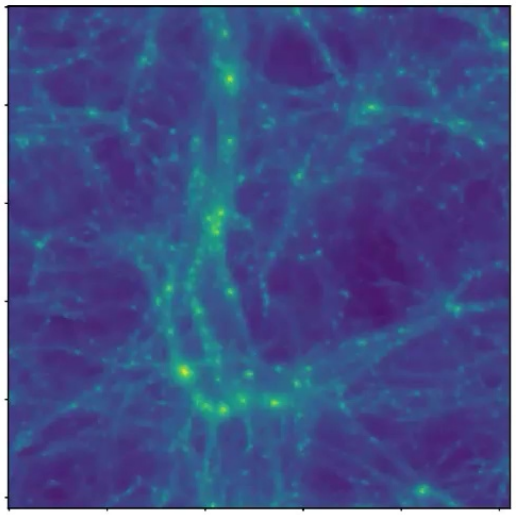
With *Francisco Villaescusa-Navarro and Uroš Seljak*

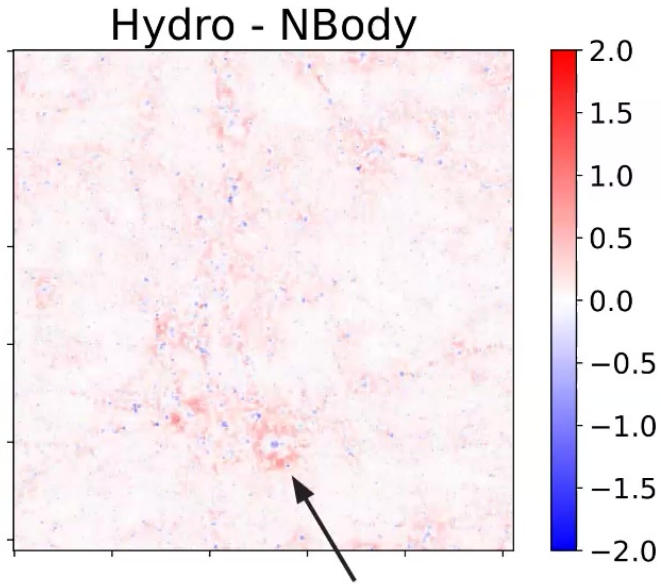


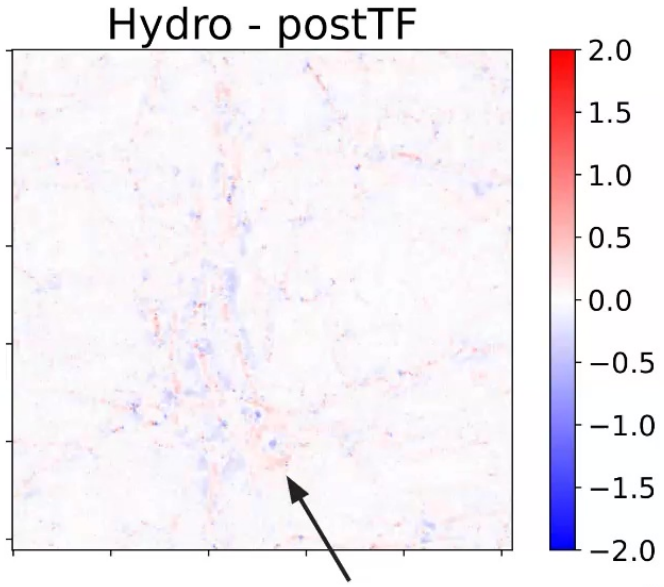
Divij Sharma

Modeling the baryonic effect as a transfer function:

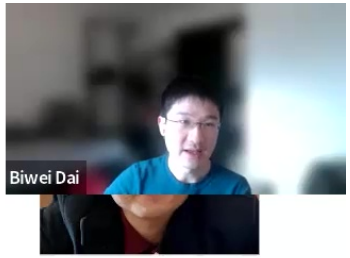
$$\delta(k) \rightarrow (P_{hydro}(k) / P_{DMO}(k))^{0.5}\, \delta(k)$$







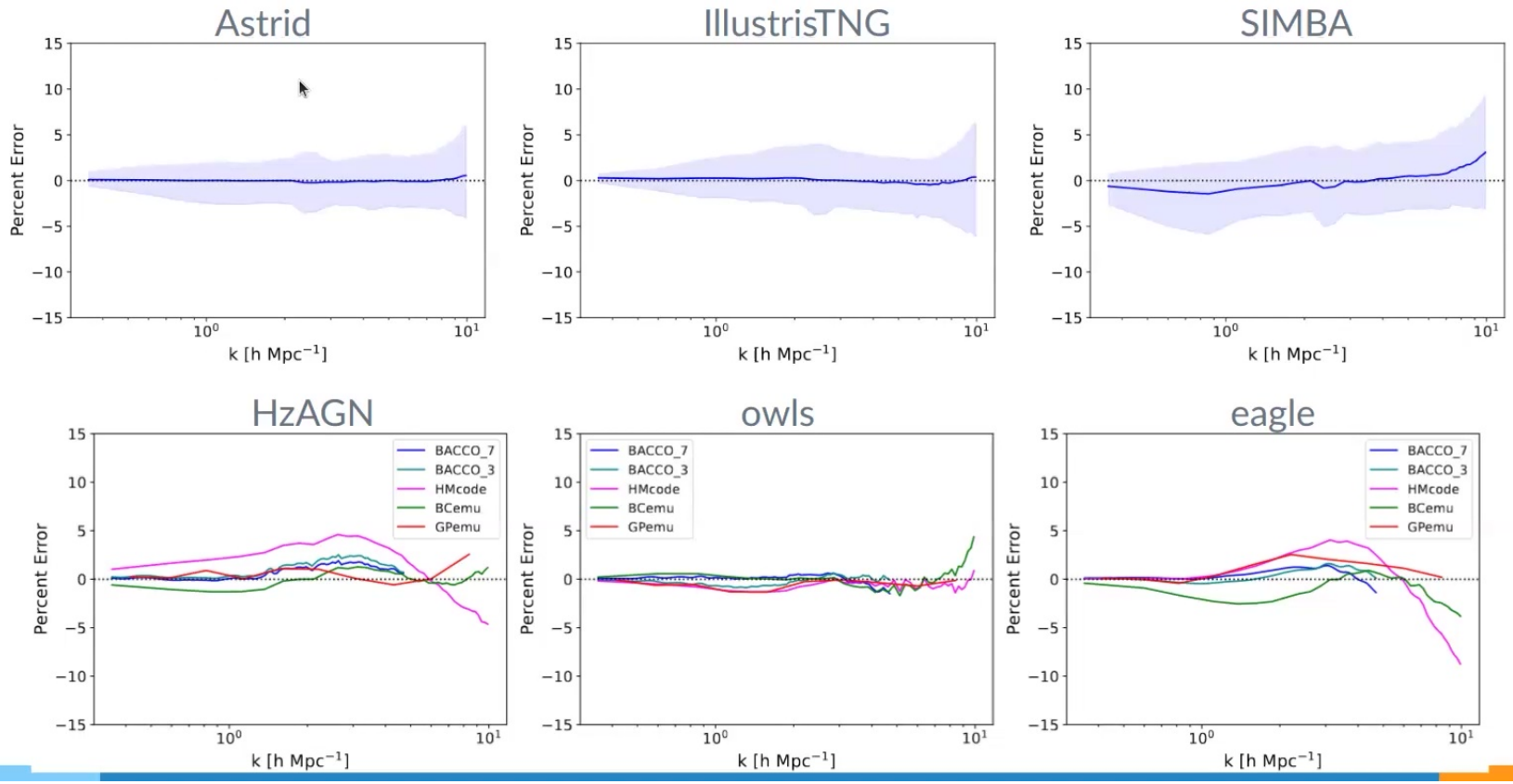
Field-level baryonic effect emulator



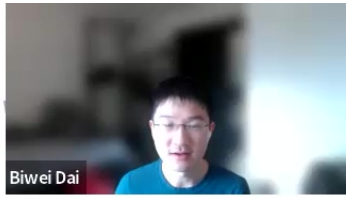
With *Francisco Villaescusa-Navarro and Uroš Seljak*

Divij Sharma

Gaussian process (4 baryon param) emulator to model the baryonic effects on power spectrum

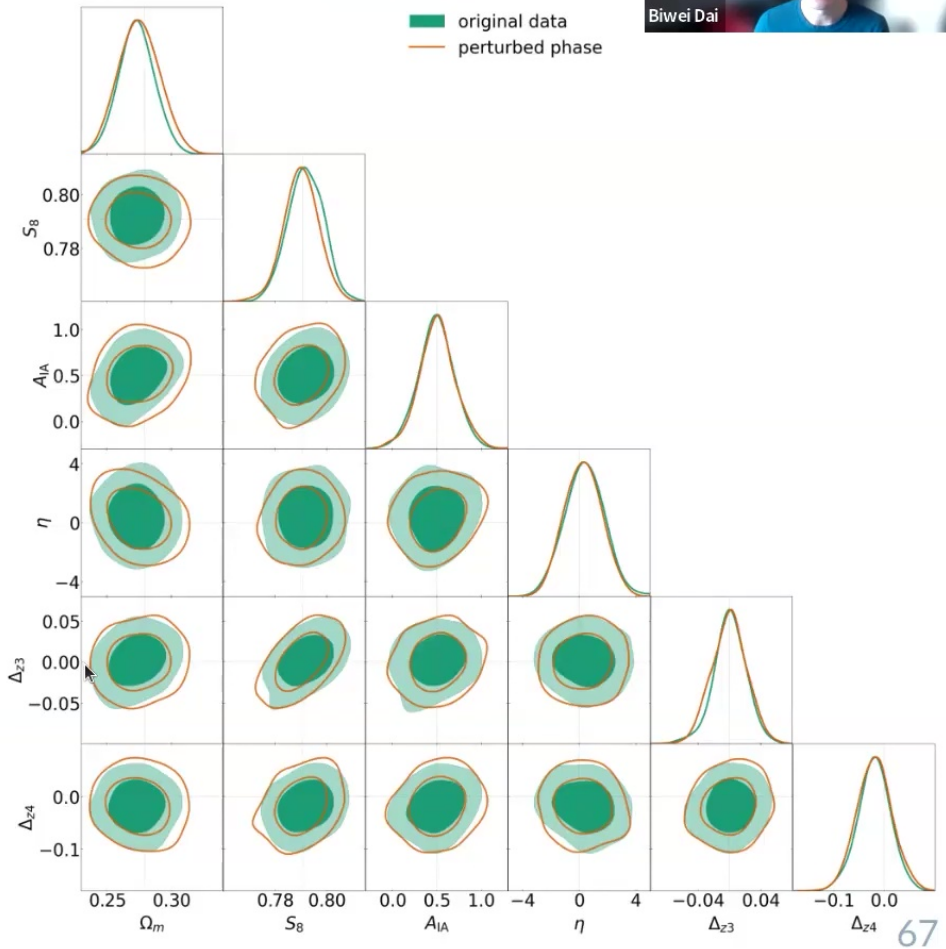


HSC cosmic shear analysis with Multiscale Flo



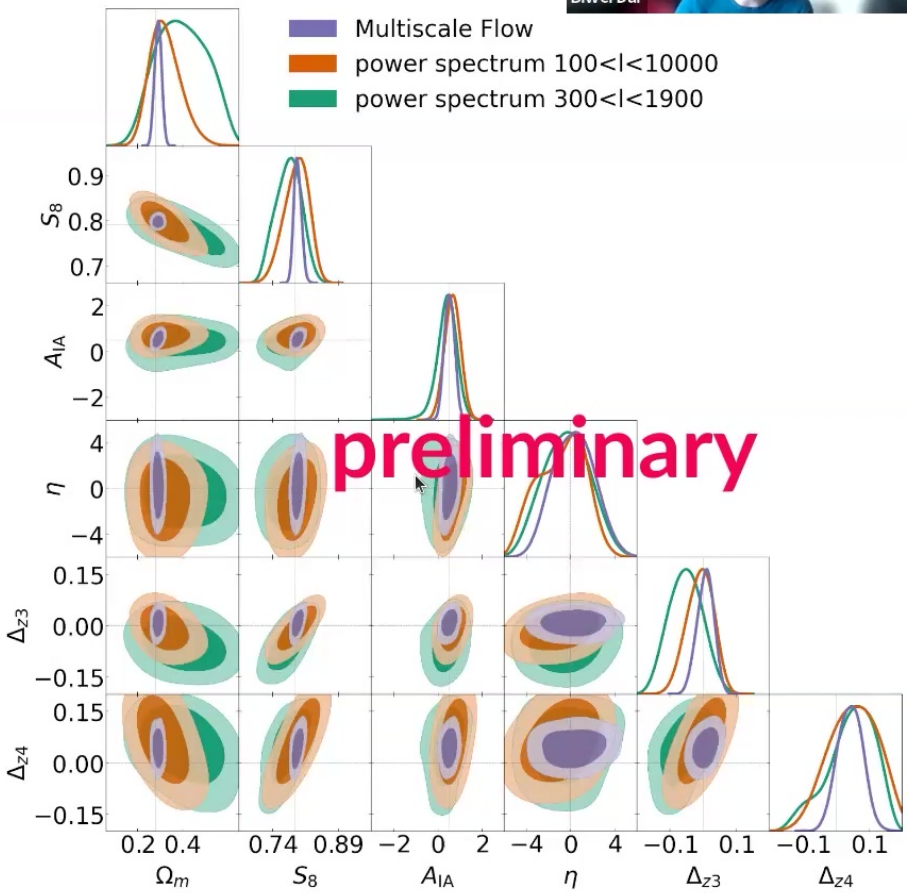
- Ignoring the phase changes due to baryonic effect doesn't bias the parameter constraints in current field-level weak lensing analysis.

- Current baryon models such as Enthalpy Gradient Descent (Dat et al. 2018) and Baryon Correction Model (radial flow, Schneider et al. 2015) couldn't improve the cross correlation coefficient in general



HSC cosmic shear analysis with Multiscale Flo

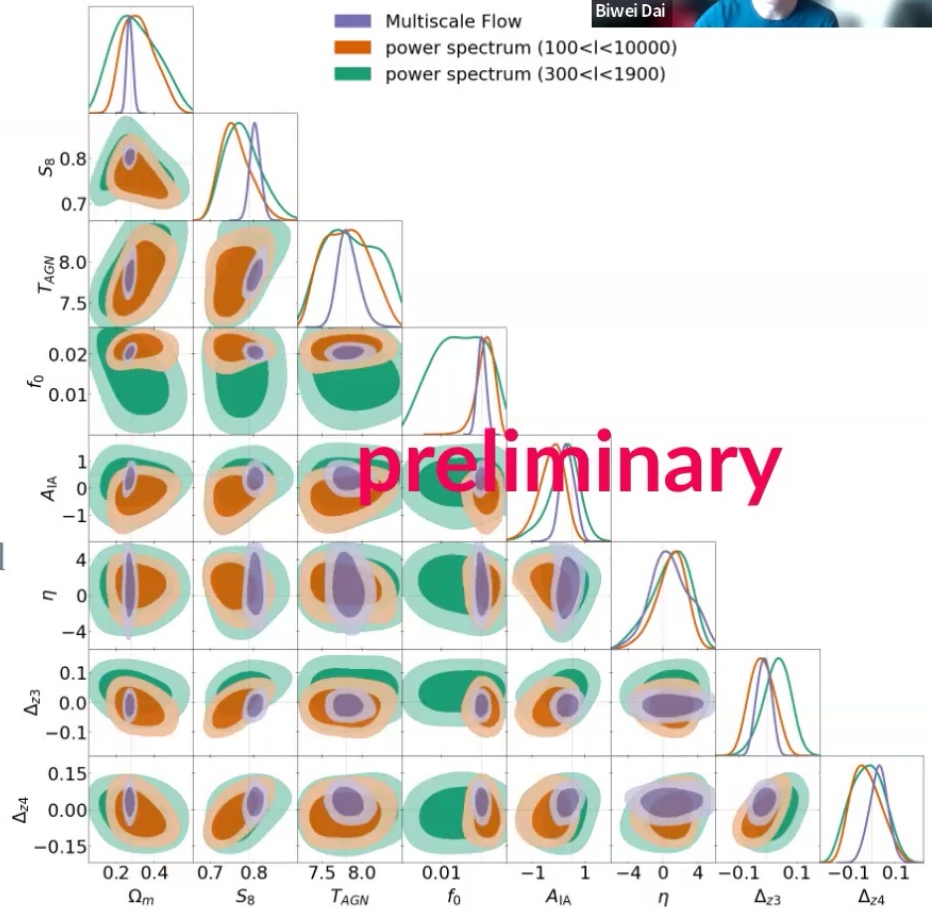
- Tests on mock data:
  - Pixel size 1.875 arcmin
  - Intrinsic alignment (filed-level NLA)
  - Photo-z uncertainty. Flat-prior on BIN 3 and BIN 4
  - Baryonic physics modeled with a transfer function



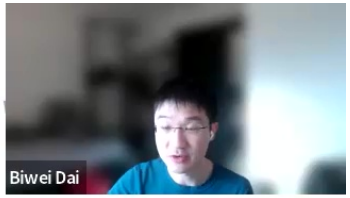
HSC cosmic shear analysis with Multiscale Flo

- Tests on mock data:
  - Pixel size 1.875 arcmin
  - Intrinsic alignment (filed-level NLA)
  - Photo-z uncertainty. Flat-prior on BIN 3 and BIN 4
  - Baryonic physics modeled with a transfer function

- About 40% degradation of constraining power after considering baryonic effects, for both Multiscale Flow and power spectrum analysis
  - Degradation is mostly on S_8

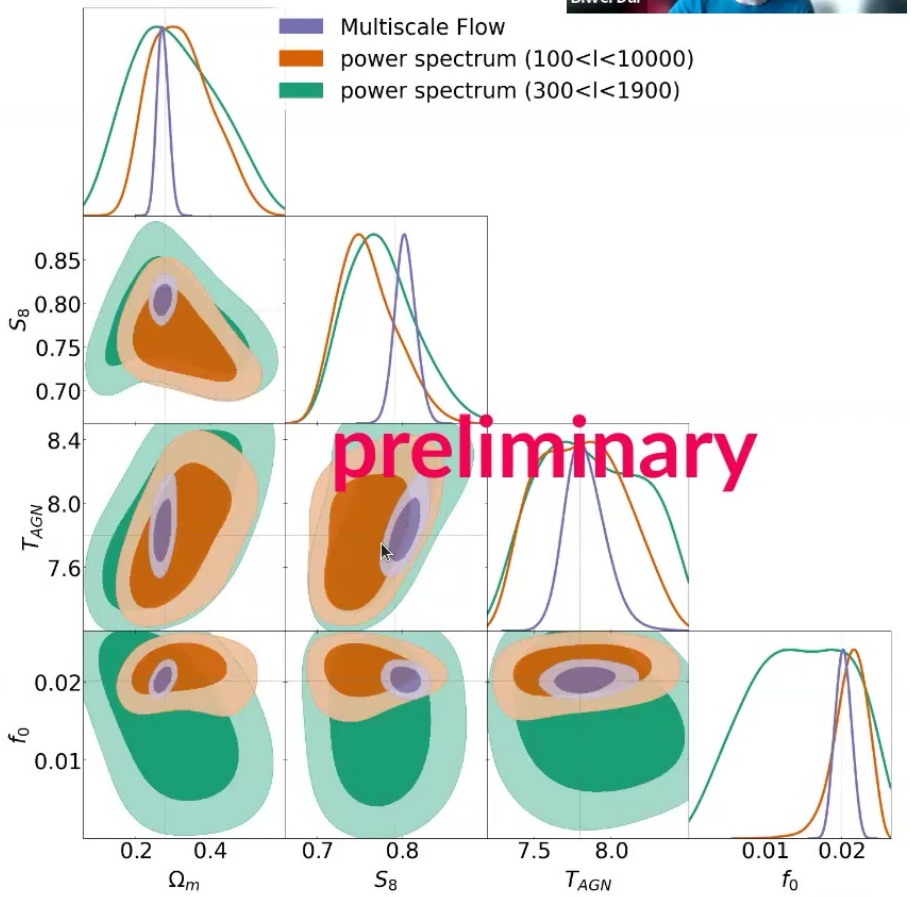


HSC cosmic shear analysis with Multiscale Flo



- Tests on mock data:
  - Pixel size 1.875 arcmin
  - Intrinsic alignment (filed-level NLA)
  - Photo-z uncertainty. Flat-prior on BIN 3 and BIN 4
  - Baryonic physics modeled with a transfer function

- About 40% degradation of constraining power after considering baryonic effects, for both Multiscale Flow and power spectrum analysis
  - Degradation is mostly on S_8
  - Strong degeneracy between S_8 and baryon parameter T_{AGN}



A Fair Universe: Unbiased Data Benchmark Ecosystem for P

Scientific Goals

Provide a large-compute-scale AI ecosystem for sharing datasets, training large models, fine-tuning those models, and hosting challenges and benchmarks.

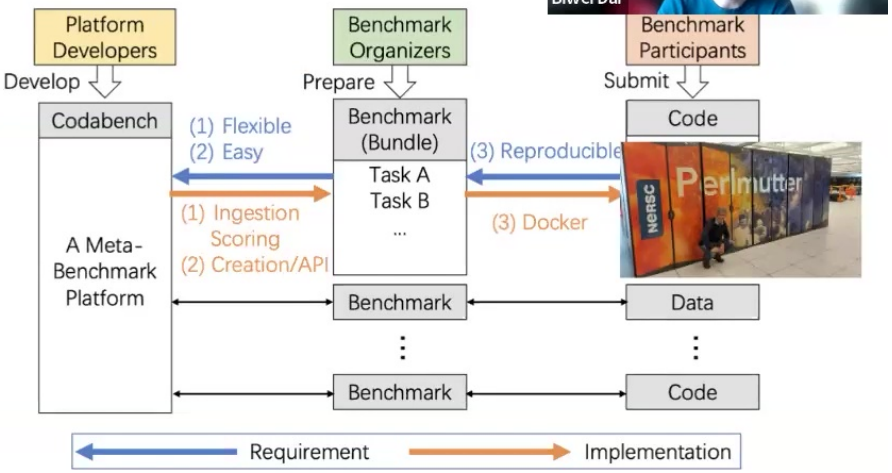
Host challenges and benchmarks focussed on discovering and minimizing the effects of systematic uncertainties.

Significance and Impact

Provide a platform for large-scale AI experimentation and development of systematic-uncertainty-aware AI models.

Research Details

- Recently funded three year comp-hep project
- Constructing datasets and tasks for challenged and *long-lived* benchmarks systematic uncertainty aware AI techniques in particle physics and cosmology
- Building HPC-enabled AI benchmark platform to host new models and be able to leverage NERSC resources to apply new AI algorithms on existing and new datasets

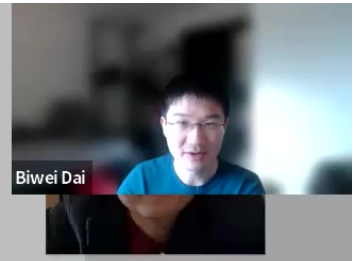


Overview of the core proposed platform (based on Codabench). Codabench is designed to support diverse benchmarks. Each benchmark is implemented by a benchmark bundle that contains one or more tasks (wrapping around datasets). This project will exploit and extend Codabench's new features; interface it to NERSC HPC capabilities and tackle the problem of systematics in physics from various angles

Conclusions

- Generative models (normalizing flows) enable Out-of-Distribution detection to improve the robustness of analysis
  - OoD detection with goodness-of-fit test
  - OoD detection with consistency of different scales
- Generative models (normalizing flows) improves interpretability by visualizing where the information is coming from, and how the constraints are made.
- Generative models (normalizing flows) are likely to extract more information with limited number of training simulations
- Physical inductive bias in normalizing flows
  - Symmetry: TRENF
  - Multiscale structure: Multiscale Flow
- We are applying the model to HSC cosmic shear analysis. Stay tuned!
- Limitations:
  - Generative models are harder and more expensive to train.
  - Working on scaling the model to higher dimensions for Rubin.

Field-level baryonic effect emulator



With *Francisco Villaescusa-Navarro and Uroš Seljak*

Divij Sharma

Gaussian process (4 baryon param) emulator to model the baryonic effects on power spectrum