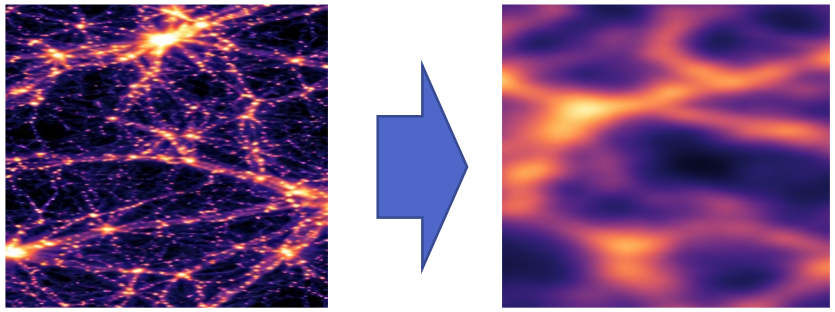
Beyond the Explicit Likelihood: Novel Methods for Parameter and Model SBI

with Jameson Dong (JHU), Carolina Cuesta-Lazaro(NYU/Flatiron),
Niall Jeffrey (KCL), Ben Wandelt (JHU),
Sven Krippendorf (Cambridge),
Jochen Weller (LMU/MPE)...

Good data needs good inference

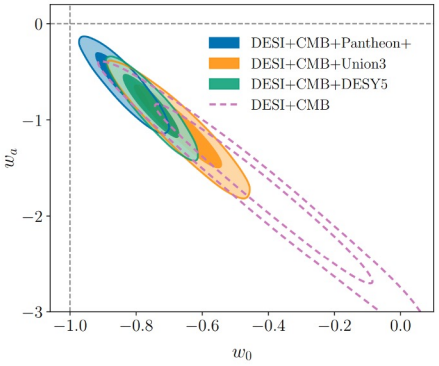


Much of our data remains unused

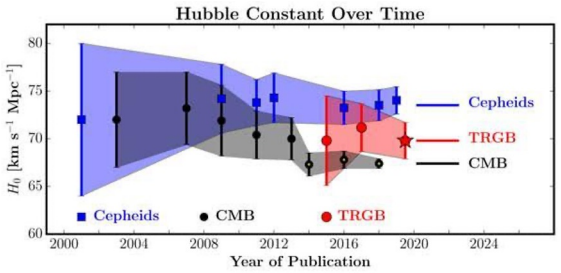
Extract more information

Parameter SBI

DESI+24



Freedman+19



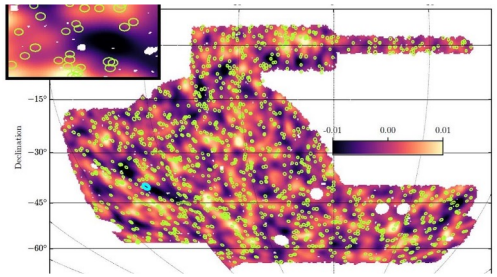
Tensions are on the rise

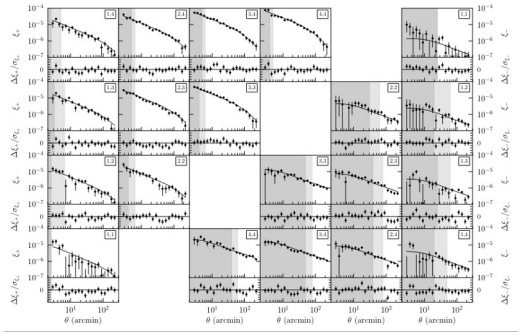
Quantify model fits and find and new models

Model SBI

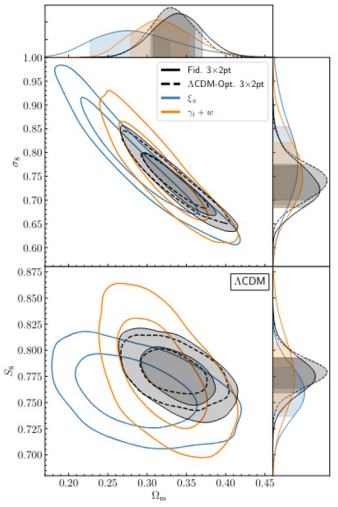
Traditional inference

Data | Summary | Likelihood | Inference





$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

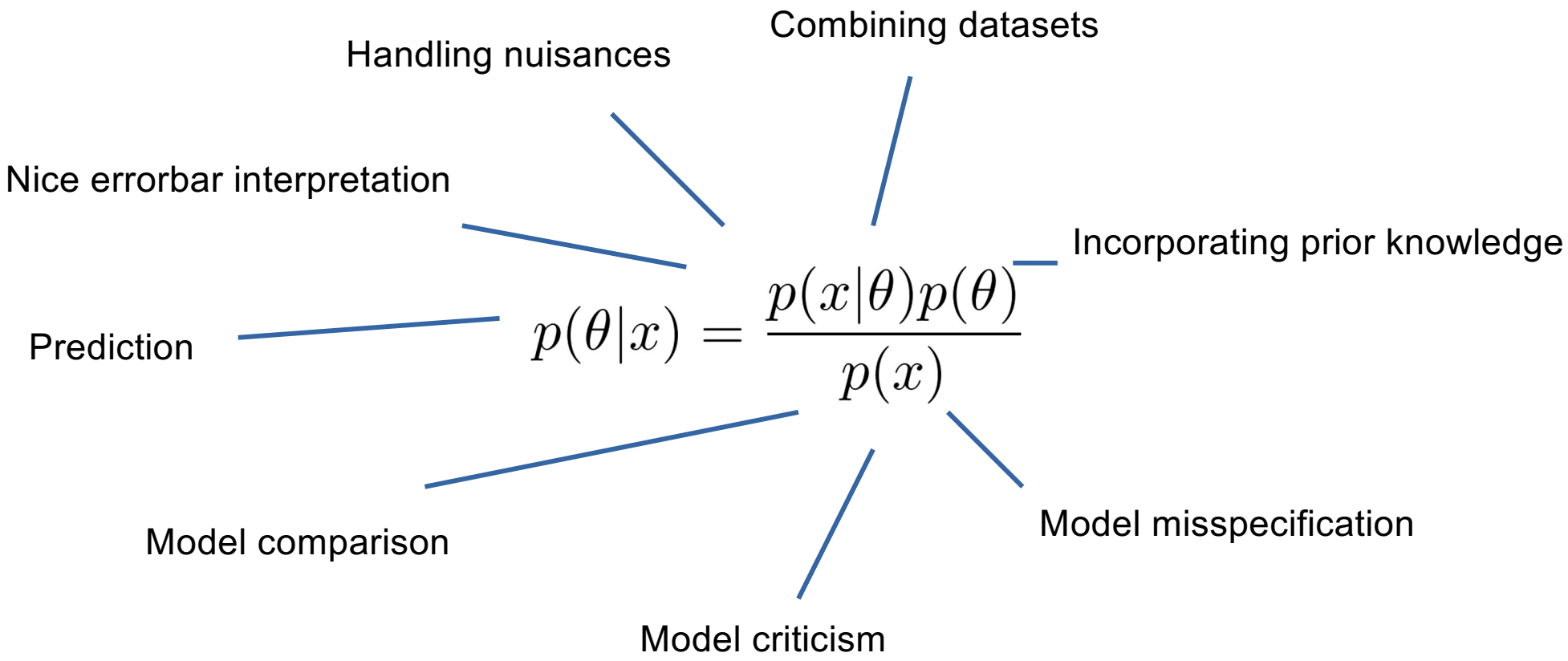


iff

- Model well specified
- Prior is available
- Likelihood is tractable

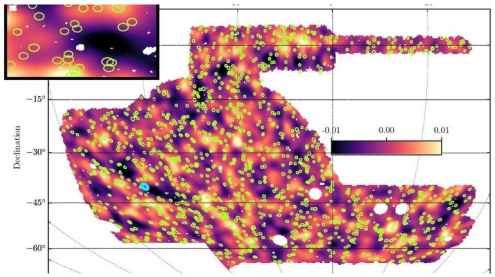
DES+ 22

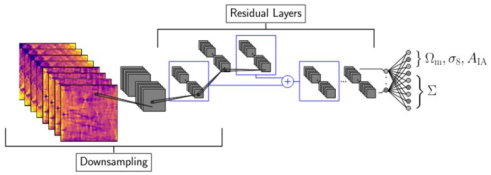
What can we do with Bayes

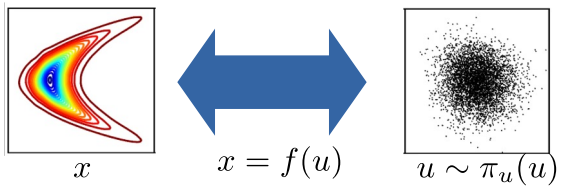


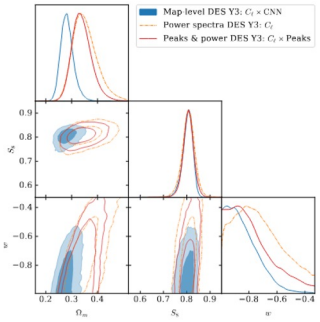
SBI is changing this

Data | Summary | Likelihood | Inference





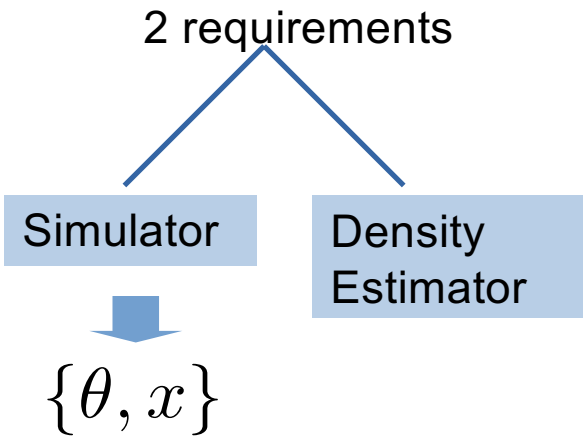




- Simulation based inference enables inference with intractable likelihoods
- Summary extraction can be optimized

DES+ 22
Fluri+ 19
Papamakarios+18

SBI for intractable likelihoods

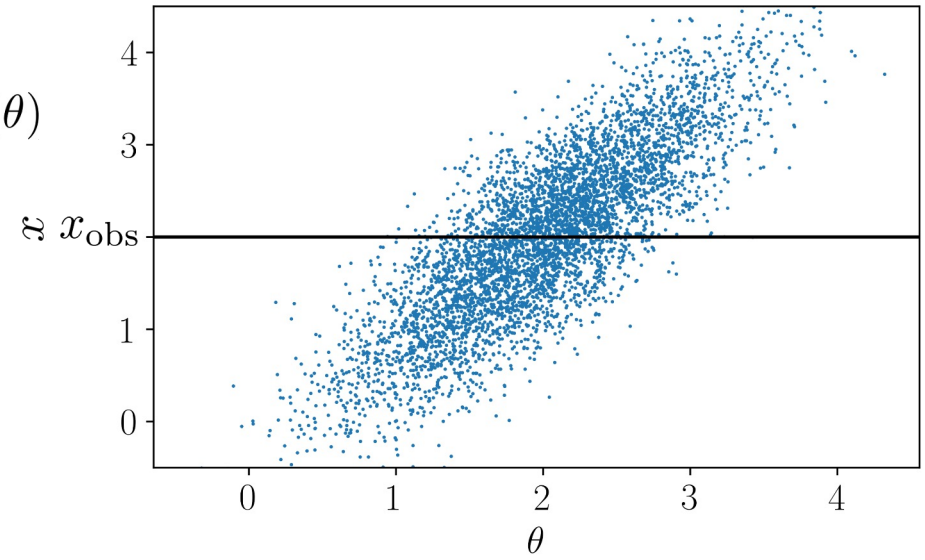


- No analytical likelihood
- No analytical covariance

Likelihood is implicitly modelled in the simulator

SBI for intractable likelihoods

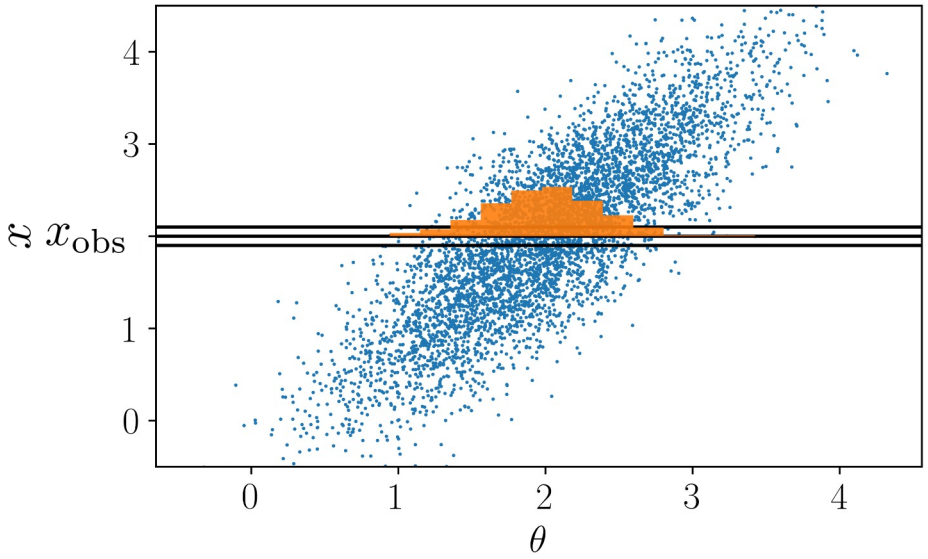
- Sample from the joint pdf $p(x, \theta) = p(x|\theta)p(\theta)$
- Approximate posterior obtained from samples in interval around observation
- Exact for vanishing intervals



SBI for intractable likelihoods

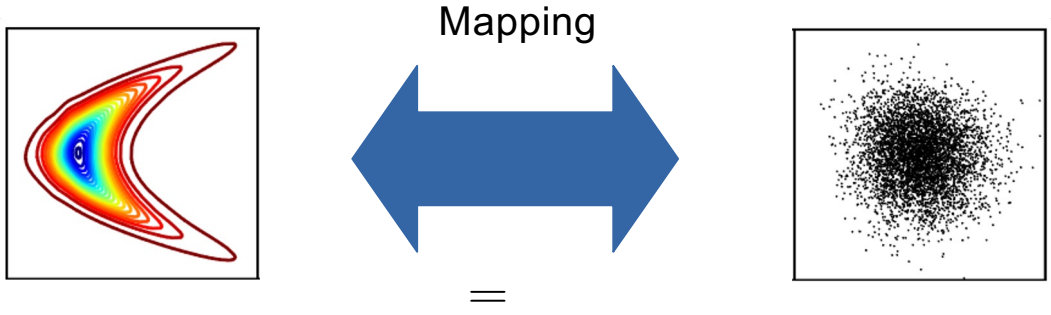
We can learn the posterior directly

$$q(\theta|x)$$



SBI with neural density estimators

Papamakarios+ 17



Change of Variables

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

Neural Density Estimation with Normalizing Flows

True distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ | Estimate $q_\varphi(\boldsymbol{\theta}|\boldsymbol{x})$

$$\begin{aligned}\mathbb{E}_{p(\boldsymbol{x})}[D_{\mathrm{KL}}(p(\boldsymbol{\theta} \mid \boldsymbol{x}) \,\|\, q_\varphi(\boldsymbol{\theta} \mid \boldsymbol{x}))] &= \int d\boldsymbol{x}\, p(\boldsymbol{x}) \int d\boldsymbol{\theta}\, p(\boldsymbol{\theta} \mid \boldsymbol{x}) \ln\left(\frac{p(\boldsymbol{\theta} \mid \boldsymbol{x})}{q_\varphi(\boldsymbol{\theta} \mid \boldsymbol{x})}\right) \\ &= -\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\ln q_\varphi(\boldsymbol{\theta} \mid \boldsymbol{x})\right] + \text{const.} \\ &\approx \underbrace{-\frac{1}{N_{\mathrm{sim}}} \sum_n \ln q_\varphi(\boldsymbol{\theta}_n \mid \boldsymbol{x}_n)}_{\text{Independent of true distribution}} + \text{const.}\end{aligned}$$

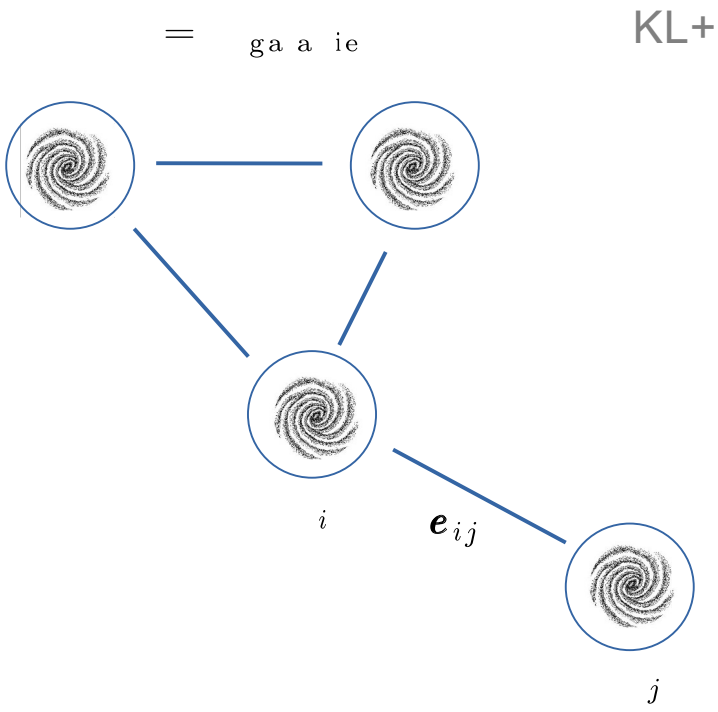
SBI maximizes the information of the summary

Summary network $h_\lambda(\boldsymbol{d})$

$$
\begin{aligned}
\mathrm{MI}(h_\lambda(\mathbf{x}), \boldsymbol{\theta}) &= D_{\mathrm{KL}}\left(p(h_\lambda(\mathbf{x}), \boldsymbol{\theta}) \parallel p(h_\lambda(\mathbf{x}))\, p(\boldsymbol{\theta})\right) \\
&= \int \mathrm{d}^n\boldsymbol{\theta}\, \mathrm{d}^n\mathbf{x}\, p(\mathbf{x}, \boldsymbol{\theta}) \log\left(\frac{p(h_\lambda(\mathbf{x}), \boldsymbol{\theta})}{p(h_\lambda(\mathbf{x}))\, p(\boldsymbol{\theta})}\right) \\
&= \int \mathrm{d}^n\boldsymbol{\theta}\, \mathrm{d}^n\mathbf{x}\, p(\mathbf{x}, \boldsymbol{\theta}) \log\left(\frac{p(\boldsymbol{\theta} \mid h_\lambda(\mathbf{x}))}{p(\boldsymbol{\theta})}\right) \\
&\approx \int \mathrm{d}^n\boldsymbol{\theta}\, \mathrm{d}^n\mathbf{x}\, p(\mathbf{x}, \boldsymbol{\theta}) \log q_\phi(\boldsymbol{\theta} \mid h_\lambda(\mathbf{x})) + \mathrm{const.} \\
&= \mathbb{E}_{p(\mathbf{x}, \boldsymbol{\theta})}\left[\log q_\phi(\boldsymbol{\theta} \mid h_\lambda(\mathbf{x}))\right] + \mathrm{const.}
\end{aligned}
$$

Architecture can be freely adapted to data

- e.g. galaxy catalogs can be represented as graphs
- Distances are encoded in the graph edges rotationally & translationally invariant
- Graph neural network can learn summary statistics to avoid lossy power spectrum compression



We can learn about our simulations

KL+ 25

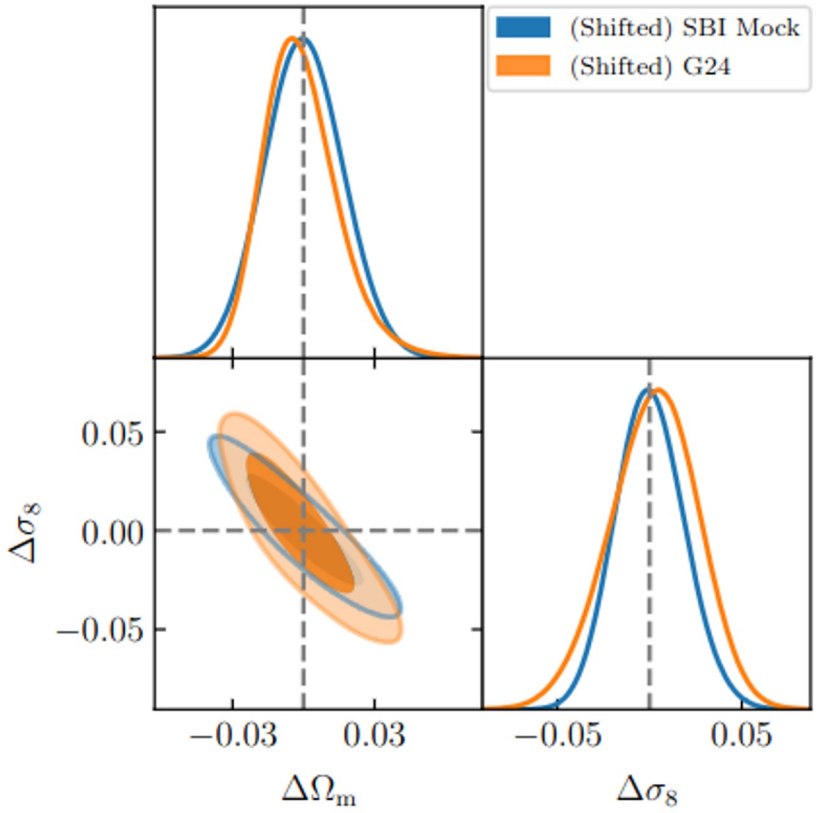
Some simulations are more flexible than others

We can identify scales of our inference problem

Architecture can be freely adapted to data

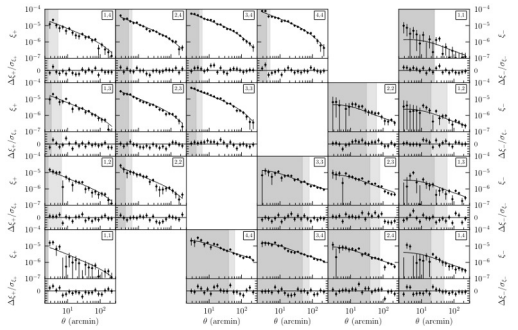
Zelmer, Bulbul, KL+ 26

- Clusters catalogs do not have an order
- We can include this permutation invariance
- EROSITA is currently applying this



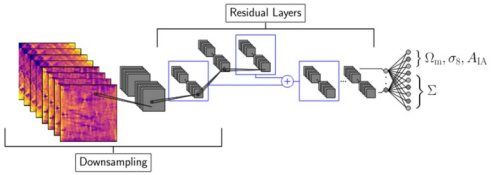
Summary statistics in cosmology

Traditional, handcrafted summaries



- Clear theoretical interpretation
- Modeling / estimation expensive
- Fails to capture significant amount of information

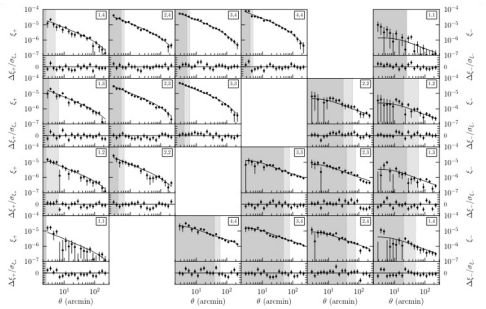
Optimized summaries



- Highly efficient in information extraction
- Cheap once simulations are run
- (Largely) fails to deliver feeling of interpretability

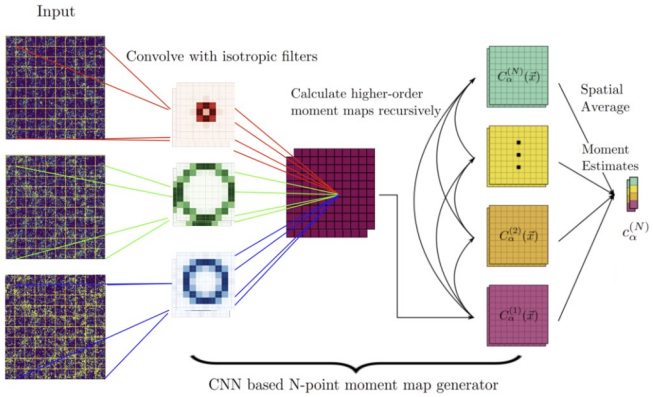
Summary statistics in cosmology

Traditional, handcrafted summaries



- Clear theoretical interpretation
- Modeling / estimation expensive
- Fails to capture significant amount of information

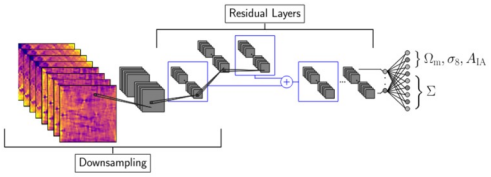
Optimized traditional summaries



- Clear theoretical interpretation
- Cheap once simulations are run
- Easily incorporate arbitrarily high orders

Miles+ 21
Gong+ 24

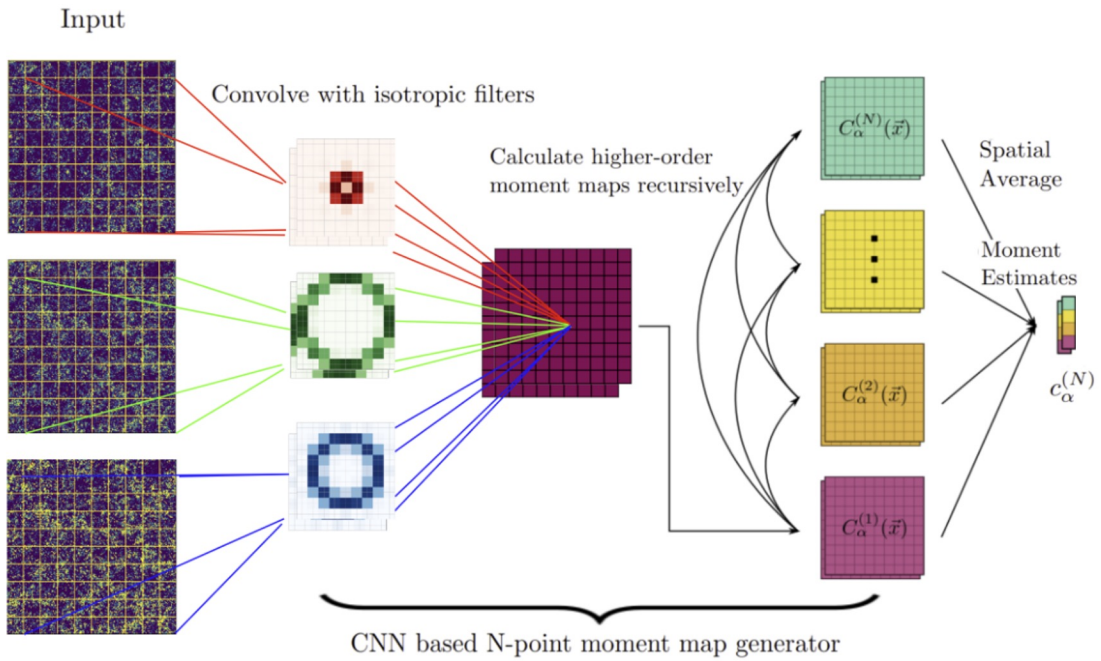
Optimized summaries



- (Largely) fails to deliver feeling of understanding
- Cheap once simulations are run
- Highly efficient in information extraction

C3NN: Cosmological Correlator Convolutional Neural Network

- C3NN is a heavily constrained one-layered CNN
- After the single convolution we deterministically construct "moment maps"
- Their spatial average can be seen as a weighted integral over real space n-point functions



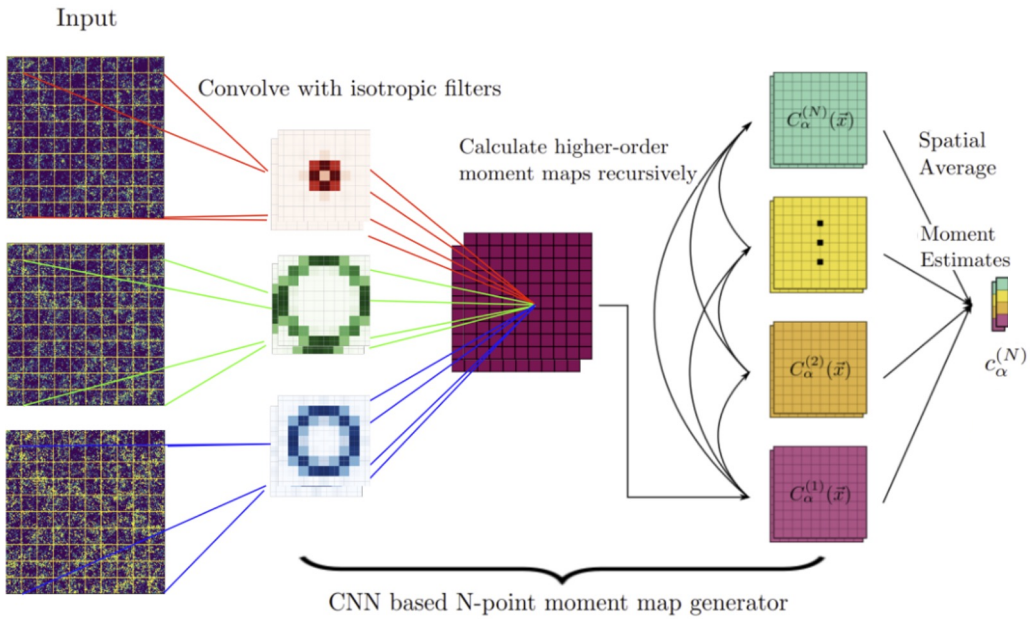
Miles+ 21
Gong+ 24

C3NN: Cosmological Correlator Convolutional Neural Network

Standard convolution

$$C_\alpha^{(1)}(\boldsymbol{x}) = \sum_{\boldsymbol{a},k} w_{\alpha,k}(\boldsymbol{a}) S_k(\boldsymbol{x}+\boldsymbol{a})$$

Sum over filter pixels, channels — weights — image



Define higher order quantities:

$$C_\alpha^{(2)}(\boldsymbol{x}) = \frac{1}{2!}\left[\sum_{(\boldsymbol{a},k)\neq(\boldsymbol{a}_1,k')} w_{\alpha,k}(\boldsymbol{a}) w_{\alpha,k'}(\boldsymbol{a}_1) S_k(\boldsymbol{x}+\boldsymbol{a}) S_{k'}(\boldsymbol{x}+\boldsymbol{a}_1)\right]$$

Moment map

$$c_\alpha^{(2)} = \frac{1}{N_{\rm pix}} \sum_{\boldsymbol{x}} C_\alpha^{(2)}(\boldsymbol{x})$$

Moment

Moments are integrated N-point estimators

Learned weights Image Pixel Values

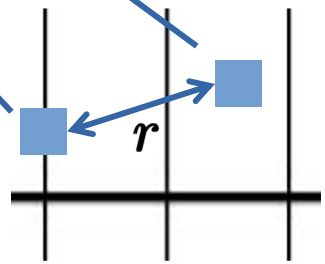
$$c_\alpha^{(2)} = \frac{1}{N_{\rm pix}} \sum_{\boldsymbol{x}} \frac{1}{2!} \left[\sum_{(\boldsymbol{a},k)\neq(\boldsymbol{a}_1,k')} w_{\alpha,k}(\boldsymbol{a}) w_{\alpha,k'}(\boldsymbol{a}_1) S_k(\boldsymbol{x}+\boldsymbol{a}) S_{k'}(\boldsymbol{x}+\boldsymbol{a}_1) \right]$$

$$= \frac{1}{2!} \sum_{(\boldsymbol{a},k)\neq(\boldsymbol{a}_1,k')} w_{\alpha,k}(\boldsymbol{a}) w_{\alpha,k'}(\boldsymbol{a}_1) \left[\frac{1}{N_{pix}} \sum_{\boldsymbol{x}} S_k(\boldsymbol{x}+\boldsymbol{a}) S_{k'}(\boldsymbol{x}+\boldsymbol{a}_1) \right]$$

output

$$= \frac{1}{2!} \sum_{(\boldsymbol{a},k)\neq(\boldsymbol{a}_1,k')} w_{\alpha,k}(\boldsymbol{a}) w_{\alpha,k'}(\boldsymbol{a}_1) \hat{\xi}_{kk'}(\boldsymbol{a}_1 - \boldsymbol{a})$$

$$= \frac{1}{2!} \sum_{(\boldsymbol{a},k)\neq(\boldsymbol{a}+\boldsymbol{r},k')} w_{\alpha,k}(\boldsymbol{a}) w_{\alpha,k'}(\boldsymbol{a}+\boldsymbol{r}) \hat{\xi}_{kk'}(\boldsymbol{r})$$



This can be done at arbitrary order!

A Note on Efficiency

- Higher orders can be defined in the same fashion

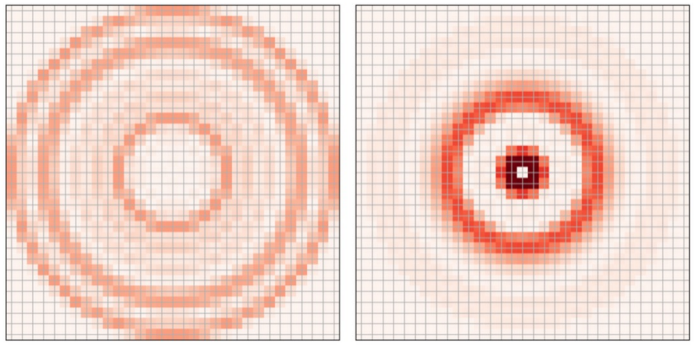
$$C_\alpha^{(N)}(\boldsymbol{x}) = \frac{1}{N!}\left[\sum_{(\boldsymbol{a},k)\neq(\boldsymbol{a}_1,k_1)\neq\ldots\neq(\boldsymbol{a}_N,k_N)}\prod_{j=1}^{N} w_{\alpha,k_j}(\boldsymbol{a}_j)S_{k_j}(\boldsymbol{x}+\boldsymbol{a})\right]$$

- Computationally efficient by use of recursion relation

$$C_\alpha^{(N)}(\boldsymbol{x}) = \frac{1}{N}\sum_{l=1}^{N}(-1)^{l-1}\left(\sum_{\boldsymbol{a},k} w_{\alpha,k}^{l}(\boldsymbol{a})S_k^l(\boldsymbol{x}+\boldsymbol{a})\right)C_\alpha^{(N-l)}(\boldsymbol{x})$$

- We enforce isotropy in our filter weights (cosmological principle)

- Some hyperparameters gain physical interpretations



Gong+ 24

Results

- Results on one withheld test cosmology for 2^{nd} – 4^{th} order

KL+26

- Higher order summaries include lower orders summaries

- Constraints tighten significantly with the inclusion of higher orders

- Roughly +20-30% for 2 -> 3 point and +10% for 3 -> 4 point for each marginal

Comparison to CNN

- We can compare to traditional CNN trained on same simulations
- Best CNN is trained on MSE first, then we run sbi
- 3k trainable parameters C3NN at fourth order vs. 165k CNN
- CNN is slightly outperforming

Validation

- How do we know that C3NN is actually picking up on higher order information?
- We delete higher order information in the convergence fields and retrain
- Fourier transform the field and reshuffle phases

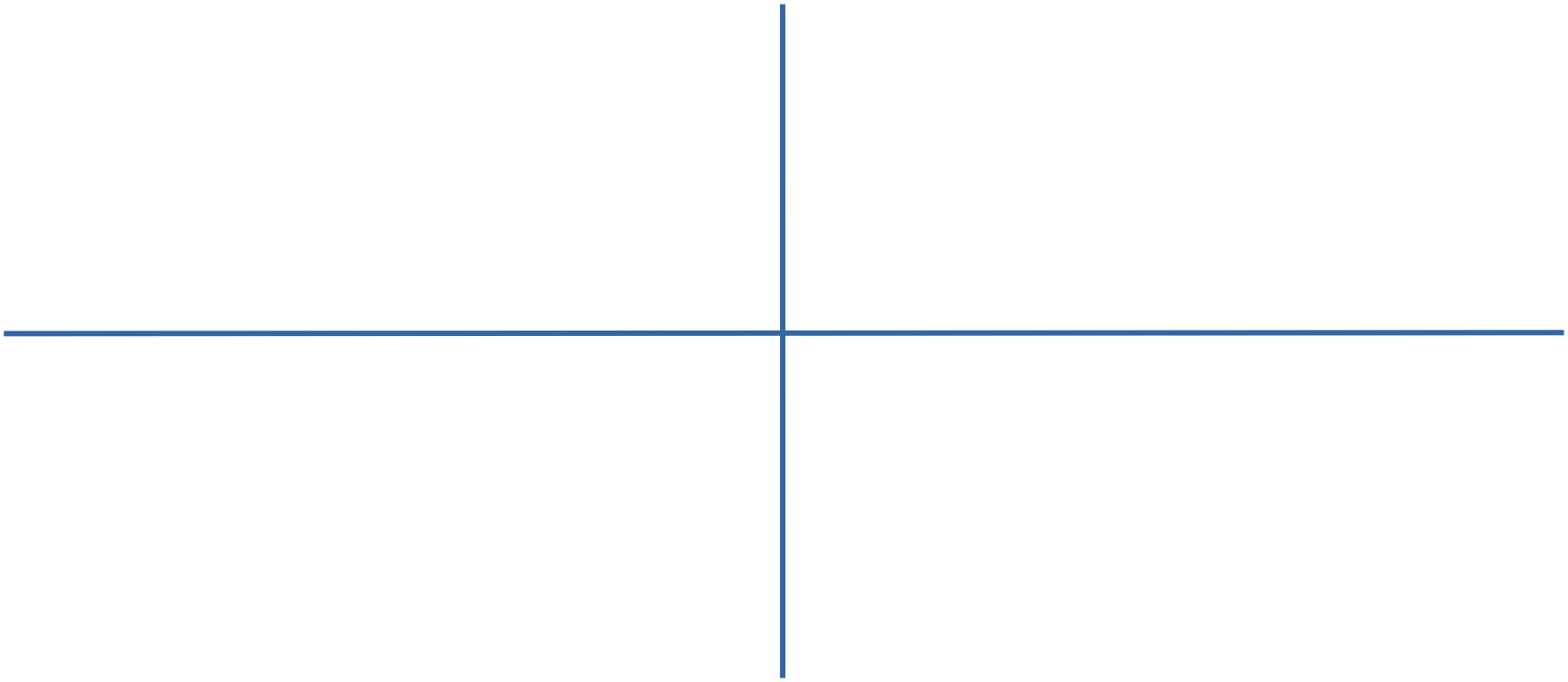
$$\delta(k) = Ae^{-i\phi}$$

- Connected higher order correlators vanish while power spectrum is preserved
- All cosmological information is now in the power spectrum / 2PCF

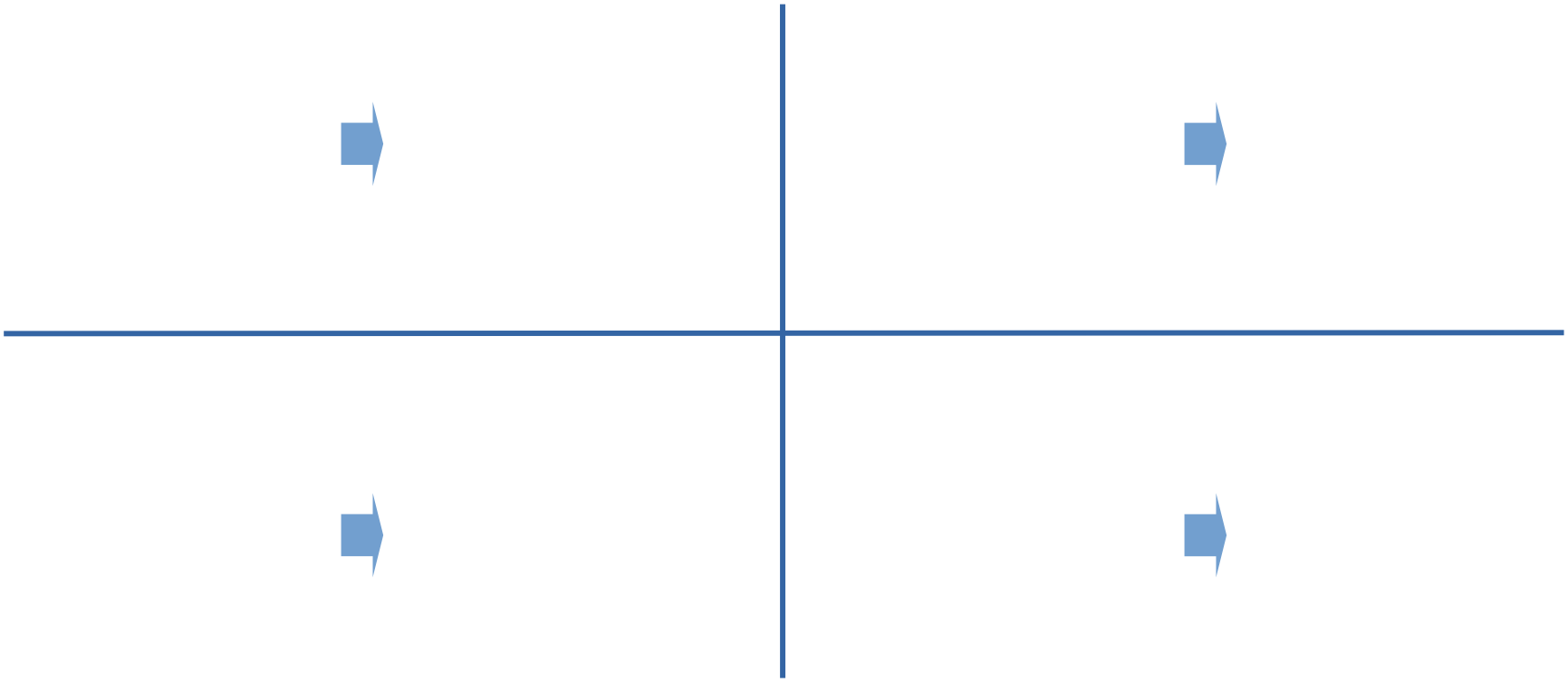


Our machine should no longer show improvements at higher orders

Validation



Validation



Phase reshuffled fields

- 2PCF encodes all cosmological information

- Trained C3NNs at all orders collapse to the same constraining power

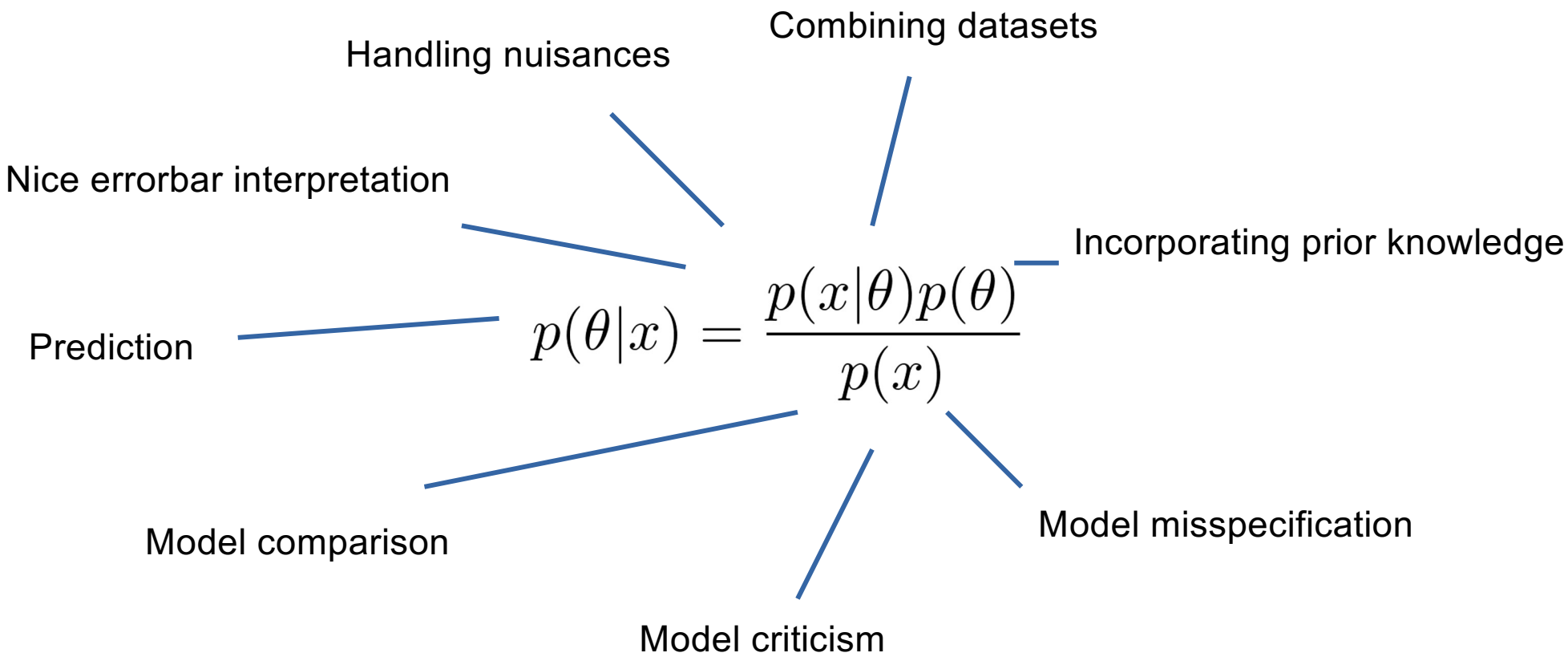
Our method does pick up on higher order n-point functions

Parameter SBI Conclusions

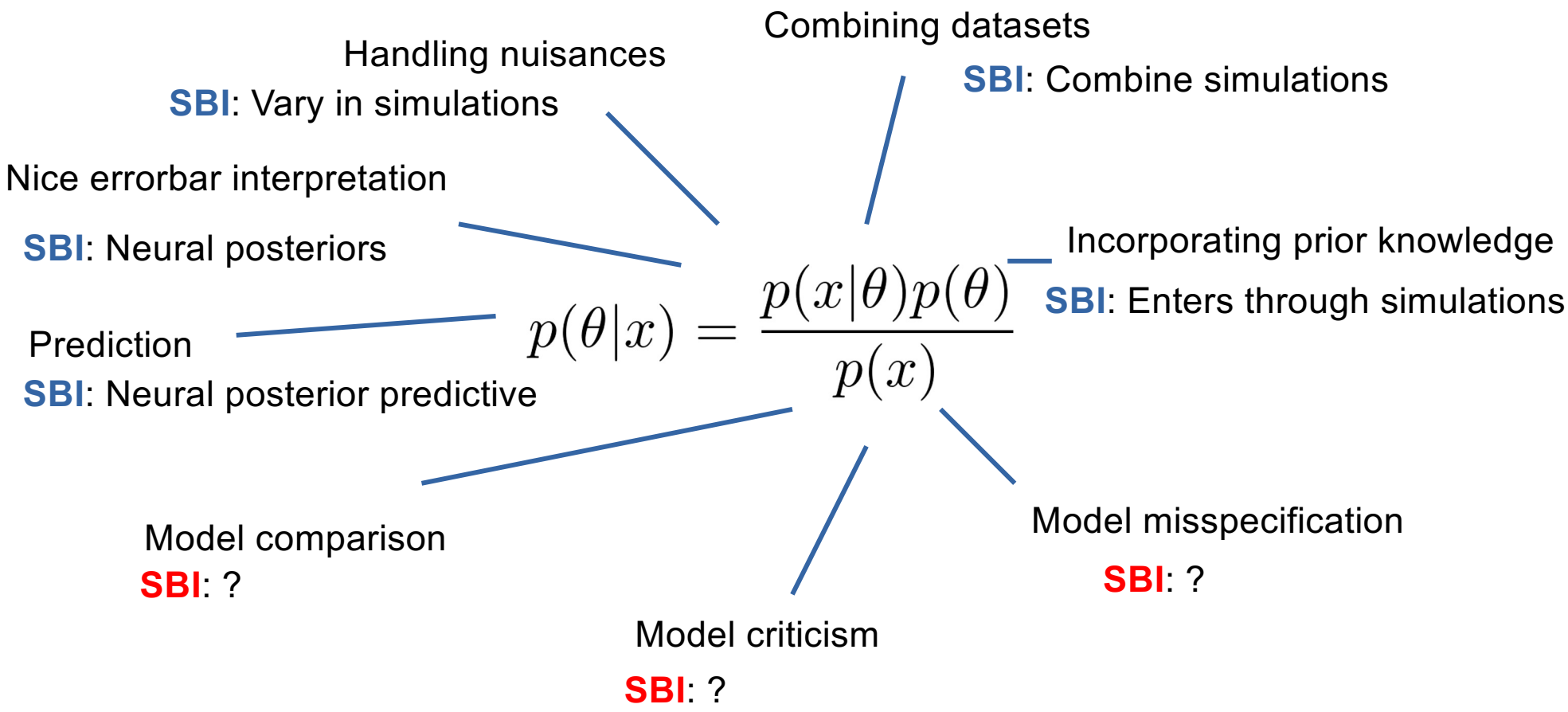
- Parameter SBI allows us to access previously inaccessible information for tight parameter constraints
- We can build architectures according to our data, e.g. graphs, sets
- We can keep some physical intuition with smart architectures

But does this tell us about the underlying physical model?

Model level SBI

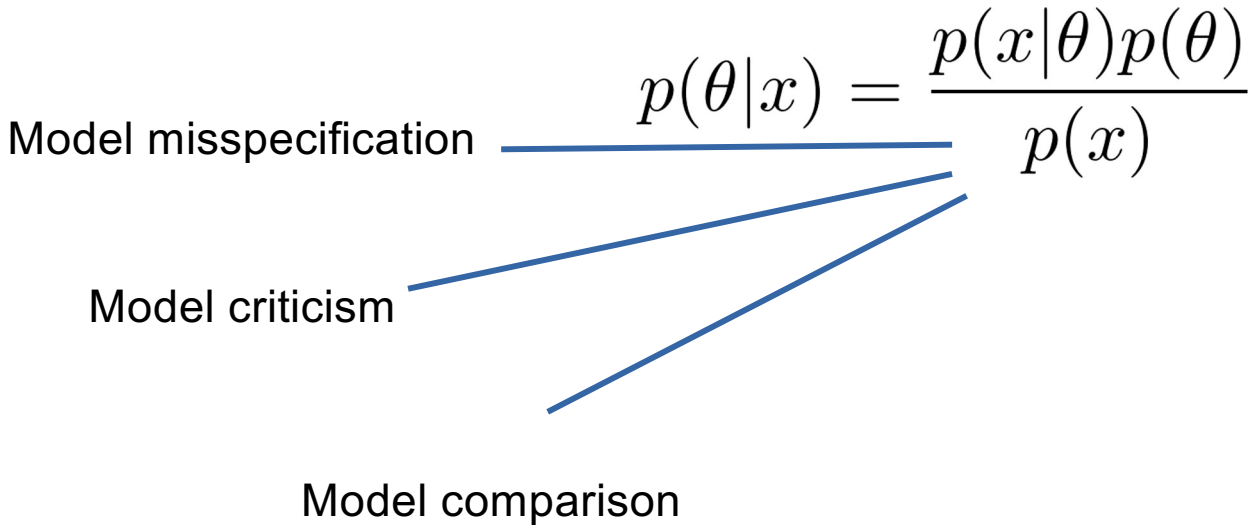


Model level SBI



What are the main criticisms of SBI today?

- How do we trust our model?
- New Physics beyond standard models?
- How do we trust the fit?



The evidence is largely overlooked in the current SBI landscape

Traditional evidences can be intractable

$$p(x|\mathcal{M}) = \int p(x|\theta, \mathcal{M}) p(\theta|\mathcal{M})\, d\theta$$

How well does the data fit my model?

- Naïve integration is usually intractable
- Amazing samplers exist (e.g. nested sampling) but require a likelihood

So what is the simulation based version of this?

Simulation-based evidences

We can sample from the data distribution, i.e. simulate:

$$\theta \sim p(\theta),\ x|\theta \sim p(x|\theta) \Rightarrow x \sim p(x)$$

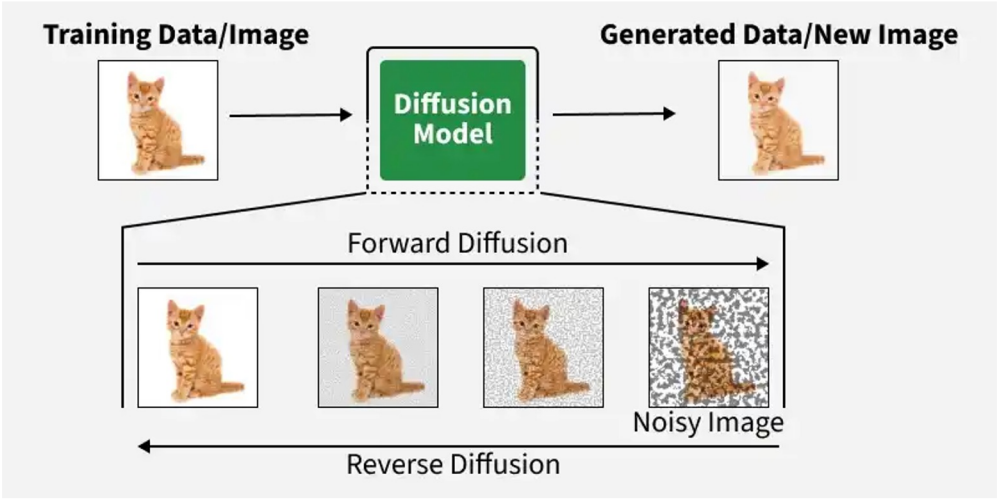
$$x \Rightarrow p(x)$$

Estimate data density given samples from the data distribution

Sounds like a traditional ML problem

$$x \Rightarrow p(x)$$





Generative models (GANs, flows, diffusion) are density estimators

e.g. Song+21

Density evaluation vs sample quality

Sample quality



$\sim p(\cdot|$  $)$

Density evaluation

$p($  $) = 1.268739 \times 10^{-20}$

Sampling quality and density quality are not the same

Evidence networks

Given data from model $\mathcal{M}_0$nd model $\mathcal{M}_1$

$$x_0 \sim p(x|\mathcal{M}_0) \quad x_1 \sim p(x|\mathcal{M}_1)$$

And labels $m \in \{0, 1\}$

$$\mathcal{L} = e^{(\frac{1}{2} - m) f(x)}$$



$$f^\star(x_O) = \log \frac{p(x_O \mid \mathcal{M}_1)}{p(x_O \mid \mathcal{M}_0)} = \log K$$

is minimized by

Jeffrey+ 24

Evidence networks learn evidence ratios

Risk $$I[f] = \int \left[e^{-\frac{1}{2} f(x)} p(x, \mathcal{M}_1) + e^{\frac{1}{2} f(x)} p(x, \mathcal{M}_0) \right] dx$$

Variation $$0 = \frac{\delta I}{\delta f(x)} = -\frac{1}{2} e^{-\frac{1}{2} f(x)} p(x, \mathcal{M}_1) + \frac{1}{2} e^{\frac{1}{2} f(x)} p(x, \mathcal{M}_0)$$

Yields Minimizer $$f^\star(x) = \log \frac{p(x \mid \mathcal{M}_1)}{p(x \mid \mathcal{M}_0)} + \log \frac{p(\mathcal{M}_1)}{p(\mathcal{M}_0)}$$

$$p(\mathcal{M}_1) = p(\mathcal{M}_0) \implies f^\star(x_O) = \log \frac{p(x_O \mid \mathcal{M}_1)}{p(x_O \mid \mathcal{M}_0)} = \log K$$

Evidence ratio minimizes this loss function

Jeffrey+ 24

Evidence networks

$$\log K_{01} = \log \frac{p(x|\mathcal{M}_0)}{p(x|\mathcal{M}_1)} \Rightarrow \log p(x|\mathcal{M}_0) = \log K_{01} + \log p(x|\mathcal{M}_1)$$



learned known

- If one density is known, the other can be recovered
- Naive guess: let's just use a Gaussian

The density chasm problem

- Gaussian reference can be far from the data
- Classification task is easy
- We do not learn the evidence accurately if the reference model is too far from the proposal

Good reference

Bad reference

ENFR: Evidence Networks with Flow matched References

Generative models:

Good at sampling, bad at density evaluation

Evidence Networks:

Can get evidences, if a good reference is available



ENFR



Use generative model to learn the reference

Use Evidence network for the evidence

ENFR: Evidence Networks with Flow matched References

$$\log p(x|\mathcal{M}_0) = \underbrace{\log p(x|\mathcal{M}_1)}_{\text{generative model}} + \underbrace{\log \frac{p(x|\mathcal{M}_0)}{p(x|\mathcal{M}_1)}}_{\text{classifier}}$$

ENFR: Evidence Networks with Flow matched References

KL+26

- Correlated GMM

- "Evidence" is the evaluated density of the GMM

$$p(x) = \frac{1}{N} \sum_{i}^{N} w_i \; \mathcal{N}(\mu_i, \sigma_i)$$

MultiENFR: Evidence Networks with Flow matched References

- We usually want to confront the data with an array of model
- Seems wasteful to train one estimator per physical model

Binary $$\mathcal{L}(x, m) = e^{(\frac{1}{2} - m)\mathcal{J}_\alpha(f_\psi(x))}$$



Multiclass $$\mathcal{L}(x, m) = \frac{1}{N-1} \sum_{i \neq m} e^{-\frac{1}{2}(\mathcal{J}_\alpha(f_{\psi,m}(x)) - \mathcal{J}_\alpha(f_{\psi,i}(x)))}$$

MultiENFR: Evidence Networks with Flow matched References

- MultiENFR learns all combination of Bayes factors and evidences at once
- Log Bayes factor estimates are additive

MultiENFR: Evidence Networks with Flow matched References

- Known likelihood application: ODE simulations

- Integrate ODE over 64 timesteps + noise

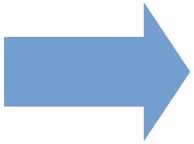
$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -2\gamma \frac{\mathrm{d}x}{\mathrm{d}t} - \omega^2 x - \beta x^3$$

ENFR outperforms standard generative models

A application to BAO data

- Simulate BAO data
- Draw Gaussian noise with DESI covariance
- 500k simulations per cosmological model (~16 hours for 5 models)
- Train ENFR (~4 hours)

Model	Likelihood-based analysis	MultiENFR	Δ
ΛCDM	-13.99±0.16	-14.07±0.34	-0.08
wCDM	-16.41±0.18	-16.00±0.35	0.40
w_0w_aCDM	-15.63±0.18	-15.46±0.40	0.17
νCDM	-14.06±0.16	-14.01±0.30	0.04
Ω_kCDM	-15.88±0.17	-15.72±0.23	0.16



Time effort ~20 hours in wall clock

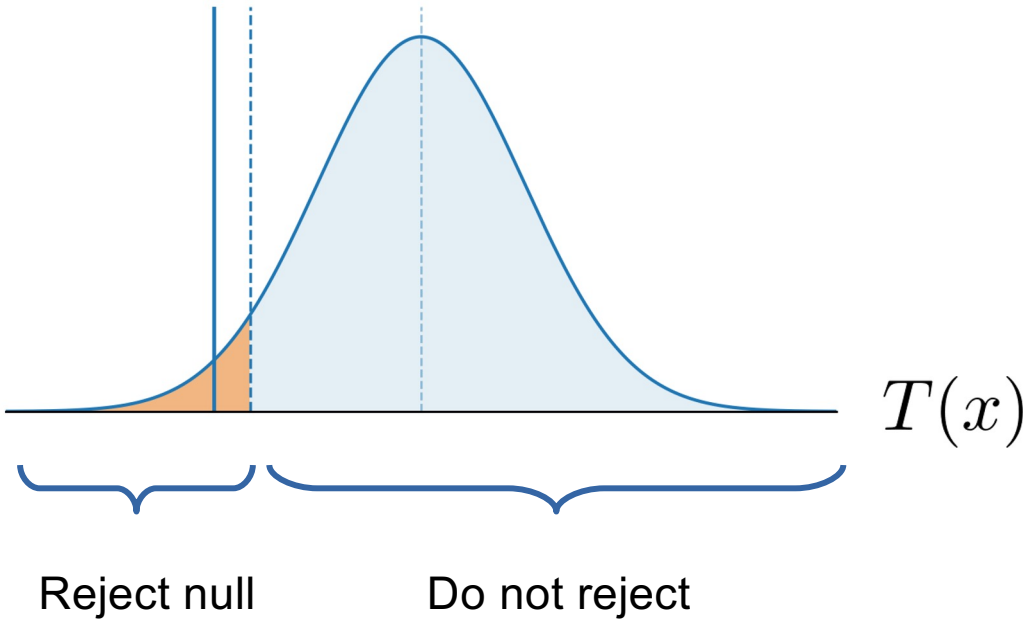
An interpretation of the evidence

- The evidence does not have an intrinsic scale

$$\log p(x|\Lambda\mathrm{CDM}) = -14.07 \; ??$$

- Hypothesis tests can reject physical models given a test statistic

- Evidence estimates are fast now once trained so we can test for a p-value



Does DESI DR2 reject LCDM?

KL+26

- Blue are evidences of 1000 LCDM simulations
- Orange is true DESI data
- Dotted line is some non-standard w(z) model
- Reminder: **20hrs** of wall clock; compare nested sampling 5 min per simulation, 5min x 1000=**3.4d** (non-parallelized)

Parameter SBI Conclusions

- SBI allows us to access previously inaccessible information
- We can build architectures according to our data, e.g. graphs, sets
- We can keep some physical intuition with smart architectures

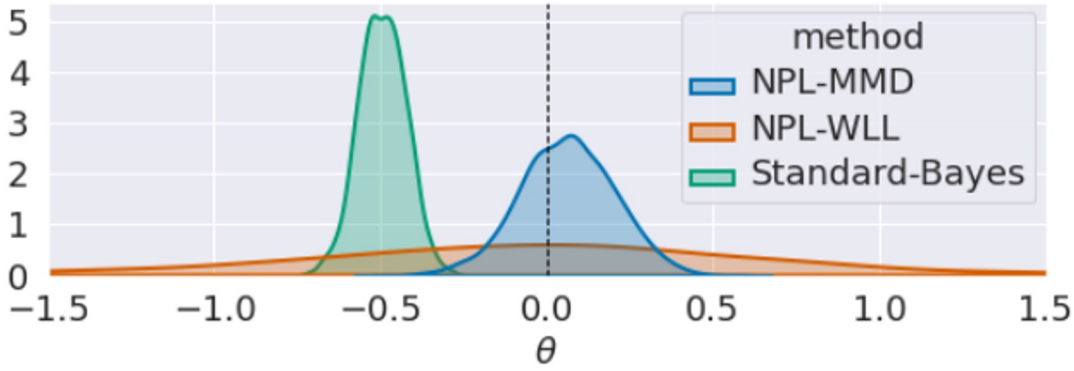
Model SBI Conclusions

- SBI should not just be about parameters
- ENFR allows for fast simulation-based evidence estimation
- We can estimate thousands of evidences instantly enabling hypothesis testing

What's missing for parameter SBI?

- Parameter inference is plagued by misspecified models
- This is not an SBI problem but a Bayes problem
- Bayesian inference is only well behaved for well-specified models
- We have to be honest about this in our inference (see General Bayes)

Grünwald+12,
Dellatporta+22



Under wrong likelihoods the posterior may not concentrate!

What's missing for model SBI?

- We can now score simulation-based models with the natural Bayesian metric
- How can we use this to discover new models? Less misspecified models?
- Evolutionary algorithms (e.g. AlphaEvolve) have shown tremendous promise for optimization problems
- Can we use LLMs to explore the model posterior?

$$p(\mathcal{M}|x) \propto p(x|\mathcal{M})p(\mathcal{M})$$