Sequential SBI for the LHC

Ranit Das

with Henning Bahl

based on arXiv:2610.XXXX

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

ML4Jets 2026

Alexander von
HUMBOLDT
STIFTUNG

SBI for the LHC

SBI for the LHC

Inferring couplings

$$|\mathcal{M}(x\,|\,\theta)|^2 = \sum_a w_a(\theta) f_a(x)$$

Polynomial structure in couplings θ

SBI for the LHC

Inferring couplings

$$|\mathcal{M}(x\,|\,\theta)|^2 = \sum_a w_a(\theta) f_a(x)$$

Polynomial structure in couplings θ

$$\log \Lambda(\theta, \theta_0) = \sum_{i=1}^{n} \log \left[\sum_a c_a(\theta)\, r_a(x_i) \right]$$

The total log-likelihood can be reconstructed from likelihood-ratio estimators trained on a **finite** set of basis processes or coupling structures.

SBI for the LHC

Inferring couplings

$$|\mathscr{M}(x\,|\,\theta)\,|^2 = \sum_a w_a(\theta) f_a(x)$$

Polynomial structure in couplings θ

$$\log \Lambda(\theta, \theta_0) = \sum_{i=1}^{n} \log \left[\sum_a c_a(\theta)\, r_a(x_i) \right]$$

The total log-likelihood can be reconstructed from likelihood-ratio estimators trained on a **finite** set of basis processes or coupling structures.

References

Brehmer et al. : arXiv:1805.12244

Brehmer et al. : arXiv:1907.10621

Brehmer et al. : arXiv:1805.00020

Brehmer et al. : arXiv:1805.00013

Brehmer et al. : arXiv:1805.12244

Bahl et al. : arXiv:2410.07315v3

Bahl et al. : arXiv:2509.05409v2

Chatterjee et al. : arXiv:2107.10859v2

Chatterjee et al. : arXiv:2205.12976v1

SBI for the LHC

Inferring couplings

$$|\mathscr{M}(x|\theta)|^2 = \sum_a w_a(\theta) f_a(x)$$

Polynomial structure in couplings θ

$$\log \Lambda(\theta, \theta_0) = \sum_{i=1}^{n} \log \left[\sum_a c_a(\theta)\, r_a(x_i) \right]$$

The total log-likelihood can be reconstructed from likelihood-ratio estimators trained on a **finite** set of basis processes or coupling structures.

Feasible since one does not need generate separate simulations for the continuous coupling values θ

References

Brehmer et al. : arXiv:1805.12244

Brehmer et al. : arXiv:1907.10621

Brehmer et al. : arXiv:1805.00020

Brehmer et al. : arXiv:1805.00013

Brehmer et al. : arXiv:1805.12244

Bahl et al. : arXiv:2410.07315v3

Bahl et al. : arXiv:2509.05409v2

Chatterjee et al. : arXiv:2107.10859v2

Chatterjee et al. : arXiv:2205.12976v1

SBI for the LHC

Inferring couplings

$$|\mathscr{M}(x|\theta)|^2 = \sum_a w_a(\theta) f_a(x)$$

Polynomial structure in couplings θ

$$\log \Lambda(\theta, \theta_0) = \sum_{i=1}^{n} \log \left[\sum_a c_a(\theta)\, r_a(x_i) \right]$$

The total log-likelihood can be reconstructed from likelihood-ratio estimators trained on a **finite** set of basis processes or coupling structures.

Feasible since one does not need generate separate simulations for the continuous coupling values θ

No need to learn parameterized likelihood ratios

References

Brehmer et al. : arXiv:1805.12244

Brehmer et al. : arXiv:1907.10621

Brehmer et al. : arXiv:1805.00020

Brehmer et al. : arXiv:1805.00013

Brehmer et al. : arXiv:1805.12244

Bahl et al. : arXiv:2410.07315v3

Bahl et al. : arXiv:2509.05409v2

Chatterjee et al. : arXiv:2107.10859v2

Chatterjee et al. : arXiv:2205.12976v1

SBI for the LHC

SBI for the LHC

Inferring mass, width

$$\left| M(x \,|\, m, \Gamma) \right|^2 \propto \frac{1}{(s - m^2)^2 + m^2\Gamma^2},$$

Dependence comes through propagators which don't have a polynomial structure.

SBI for the LHC

Inferring mass, width

$$\left| M(x \,|\, m, \Gamma) \right|^2 \propto \frac{1}{(s - m^2)^2 + m^2\Gamma^2},$$

Dependence comes through propagators which don't have a polynomial structure.

$$r(x \,|\, \theta) = \frac{p_S(x \,|\, \theta)}{p_B(x)}, \qquad \theta = (m_1, \Gamma_1, m_2, \Gamma_2, \ldots)$$

Need to learn a parameterized likelihood-ratio from simulations covering this continuous parameter space.

SBI for the LHC

Inferring mass, width

$$\left| M(x \mid m, \Gamma) \right|^2 \propto \frac{1}{(s - m^2)^2 + m^2\Gamma^2},$$

Dependence comes through propagators which don't have a polynomial structure.

$$r(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}, \qquad \theta = (m_1, \Gamma_1, m_2, \Gamma_2, \ldots)$$

Need to learn a parameterized likelihood-ratio from simulations covering this continuous parameter space.

References

- **arXiv:2002.04699: Resonance Searches with Machine Learned Likelihood Ratios**
- **arXiv:2506.00119: Generator Based Inference (GBI)**
- **arXiv:2506.06438: Data-Driven High-Dimensional Statistical Inference with Generative Models**

SBI for the LHC

Inferring mass, width

$$\left| M(x \,|\, m, \Gamma) \right|^2 \propto \frac{1}{(s - m^2)^2 + m^2\Gamma^2},$$

$$r(x \,|\, \theta) = \frac{p_S(x \,|\, \theta)}{p_B(x)}, \qquad \theta = (m_1, \Gamma_1, m_2, \Gamma_2, \ldots)$$

SBI for the LHC

Inferring mass, width

$$\left| M(x \mid m, \Gamma) \right|^2 \propto \frac{1}{(s - m^2)^2 + m^2\Gamma^2},$$

$$r(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}, \qquad \theta = (m_1, \Gamma_1, m_2, \Gamma_2, \ldots)$$

Generating simulations over a continuous parameter space can become a significant bottleneck:

- expensive detector simulations
- higher-order corrections
- low efficiencies after cuts
- Difficulty in inferring s (reconstructing the invariant mass)

Sequential SBI

Sequential SBI

Start with simulations from a few parameters from a proposal distribution $q(\theta)$

Sequential SBI

Start with simulations from a few parameters from a proposal distribution $q(\theta)$

→

Train a conditional classifier on the simulation

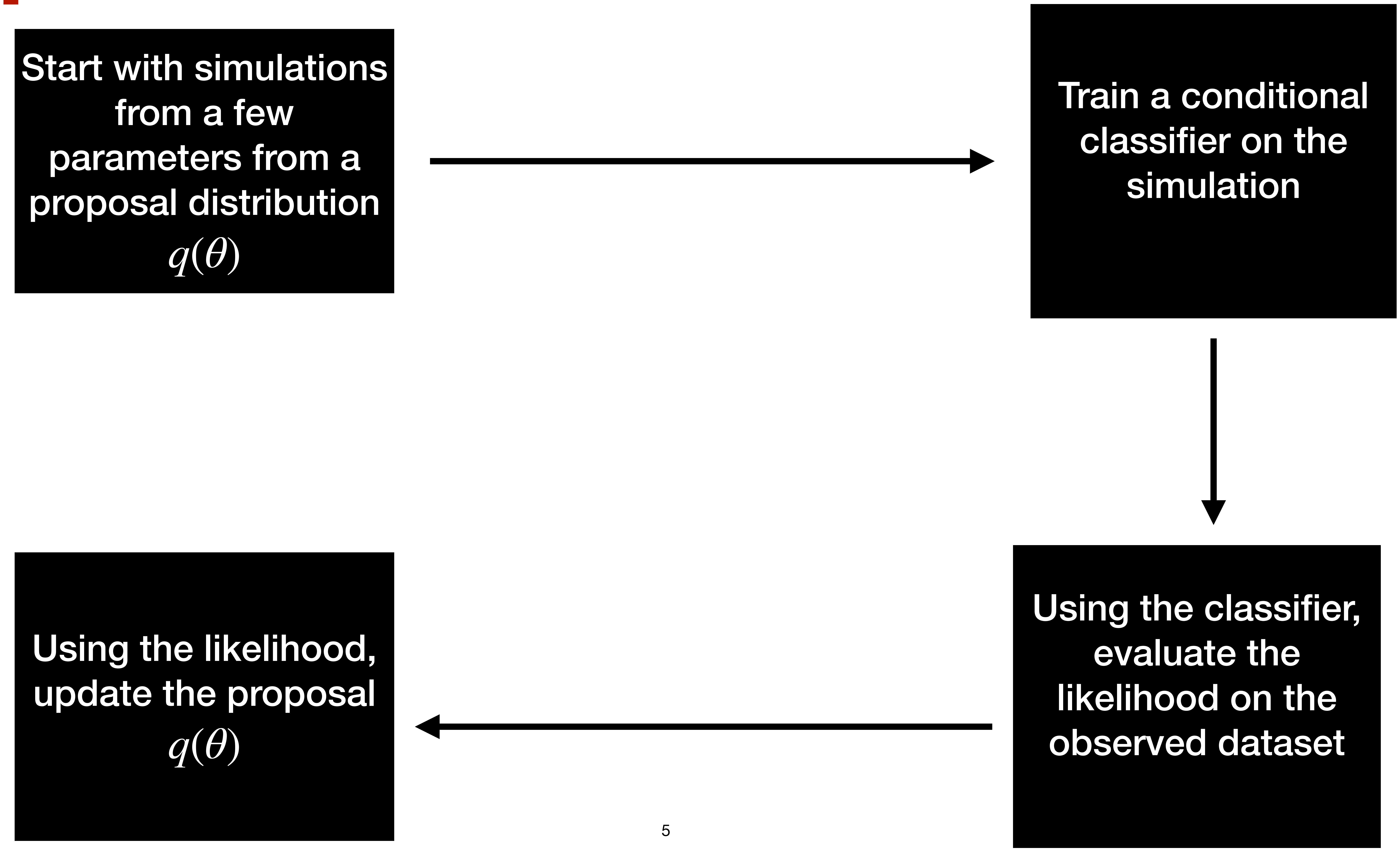
Sequential SBI

Start with simulations from a few parameters from a proposal distribution $q(\theta)$

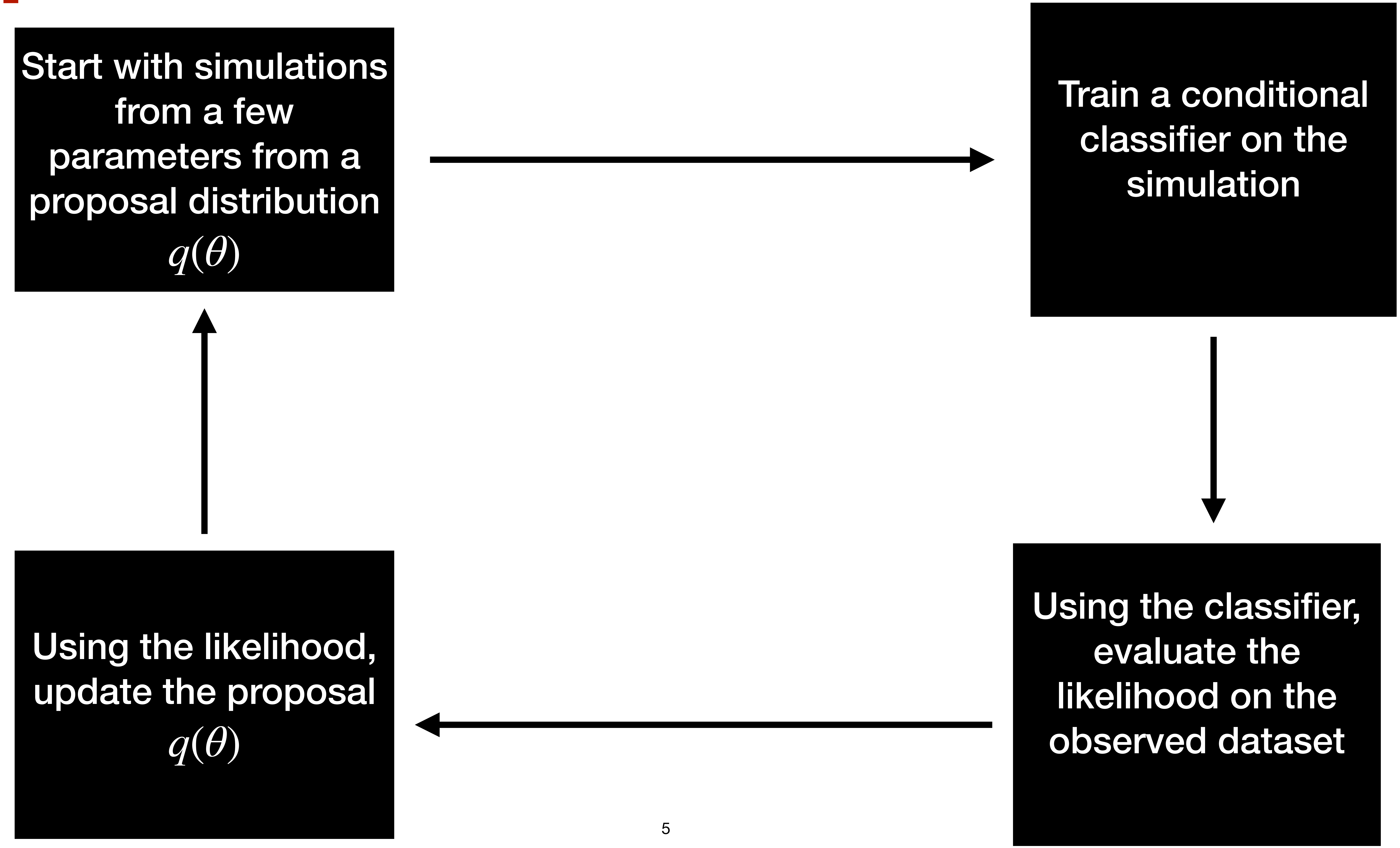
Train a conditional classifier on the simulation

Using the classifier, evaluate the likelihood on the observed dataset

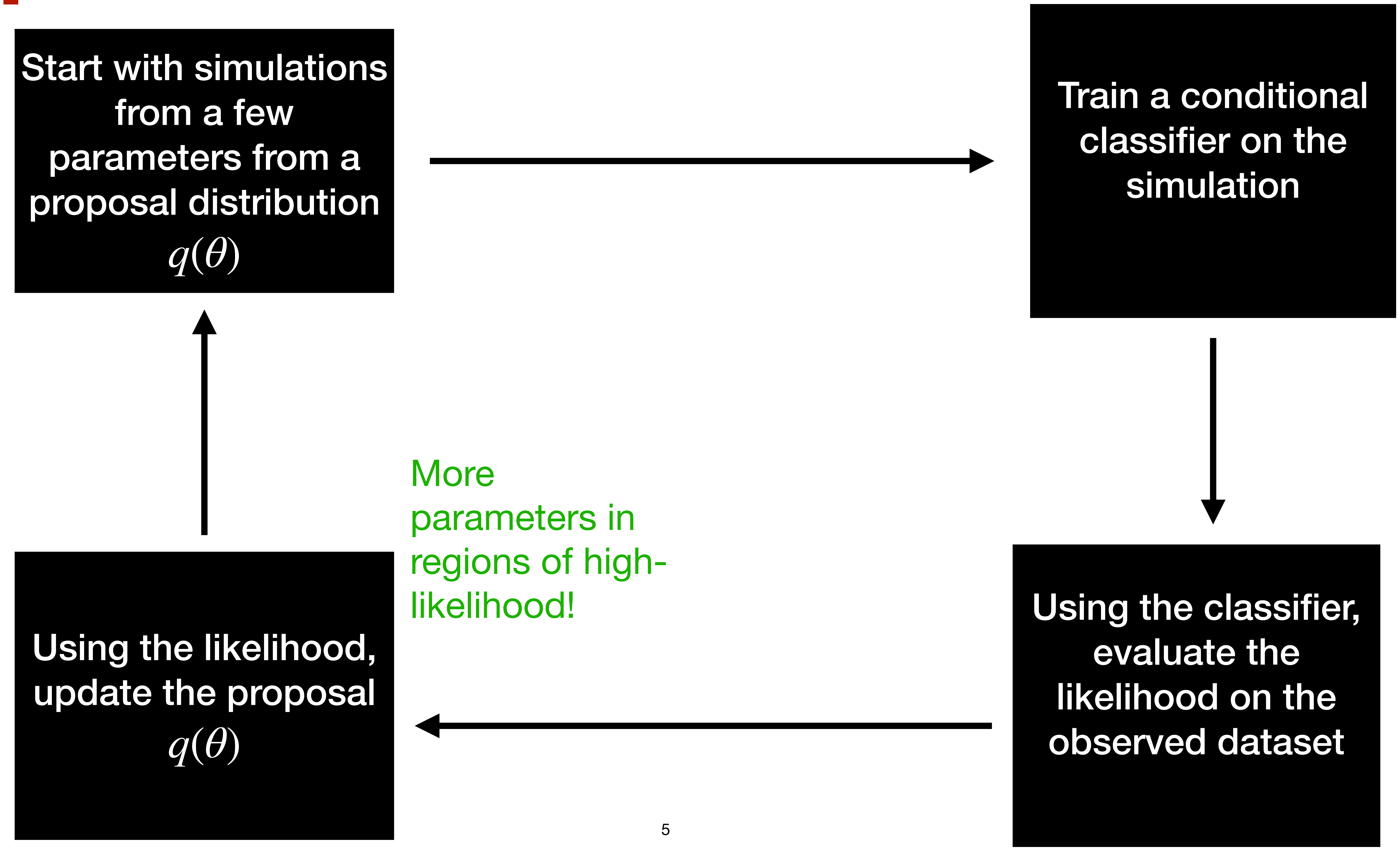
Sequential SBI



Sequential SBI



Sequential SBI



Sequential SBI

Sequential SBI

$q_{base}(\theta)$

Start with a base distribution as the proposal

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

$$r^{(t)}(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}$$

A conditional classifier

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

$$r^{(t)}(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}$$

A conditional classifier

$$\log \Lambda_{\text{shape}}(\theta, B) = \sum_{i=1}^{n} \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B} r(x_i \mid \theta)\right] - n \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B}\right]$$

Evaluate the likelihood on the observed dataset

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

$$r^{(t)}(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}$$

A conditional classifier

$$\log \Lambda_{\text{shape}}(\theta, B) = \sum_{i=1}^{n} \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B} r(x_i \mid \theta)\right] - n \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B}\right]$$

Evaluate the likelihood on the observed dataset

$$q_{t+1}(\theta) \propto q_{\text{base}}(\theta) \left[\Lambda^{(t)}(\theta)\right]^{\gamma}$$

Update proposal

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

$$r^{(t)}(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}$$

A conditional classifier

$$\log \Lambda_{\text{shape}}(\theta, B) = \sum_{i=1}^{n} \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B} r(x_i \mid \theta)\right] - n \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B}\right]$$

Evaluate the likelihood on the observed dataset

$$q_{t+1}(\theta) \propto q_{\text{base}}(\theta)\left[\Lambda^{(t)}(\theta)\right]^{\gamma}$$

Update proposal

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

$$r^{(t)}(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}$$

A conditional classifier

$$\log \Lambda_{\text{shape}}(\theta, B) = \sum_{i=1}^{n} \log\left[1 + \frac{\sigma_S(\theta)}{\sigma_B} r(x_i \mid \theta)\right] - n \log\left[1 + \frac{\sigma_S(\theta)}{\sigma_B}\right]$$

Evaluate the likelihood on the observed dataset

$$q_{t+1}(\theta) \propto q_{\text{base}}(\theta)\left[\Lambda^{(t)}(\theta)\right]^{\gamma}$$

Update proposal

With iterations, the proposal focuses on higher likelihood regions

Sequential SBI

$$q_{base}(\theta)$$

Start with a base distribution as the proposal

$$\theta_i^{(t)} \sim q_t(\theta)$$

$$x \sim p_S(x \mid \theta_i^{(t)})$$

Simulations from a proposal distribution

$$r^{(t)}(x \mid \theta) = \frac{p_S(x \mid \theta)}{p_B(x)}$$

A conditional classifier

$$\log \Lambda_{\text{shape}}(\theta, B) = \sum_{i=1}^{n} \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B} r(x_i \mid \theta)\right] - n \log \left[1 + \frac{\sigma_S(\theta)}{\sigma_B}\right]$$

Evaluate the likelihood on the observed dataset

$$q_{t+1}(\theta) \propto q_{\text{base}}(\theta) \left[\Lambda^{(t)}(\theta)\right]^{\gamma}$$

Update proposal

0<γ<1 encourages exploration beyond the high likelihood regions

With iterations, the proposal focuses on higher likelihood regions

Sequential SBI

References

arXiv:1605.06376 : Fast ε-free Inference of Simulation Models with Bayesian Conditional Density Estimation

arXiv:1905.07488 : Automatic Posterior Transformation for Likelihood-Free Inference

arXiv:1805.07226 : Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows

arXiv:2002.03712 : On Contrastive Learning for Likelihood-free Inference

arXiv:2107.01214 : Truncated Marginal Neural Ratio Estimation

arXiv:1903.00007 : Fast likelihood-free cosmology with neural density estimators and active learning

arXiv:2111.08030 : Fast and Credible Likelihood-Free Cosmology with Truncated Marginal Neural Ratio Estimation

arXiv:2209.06733 : SICRET: Supernova Ia Cosmology with truncated marginal neural Ratio EsTimation

arXiv:2304.02035 : Peregrine: Sequential simulation-based inference for gravitational wave signals

Toy example: Inference at parton level

$$r_{\mathrm{ME}}(x\,|\,\theta) = \frac{|\,\mathscr{M}_S(x\,|\,\theta)\,|^2}{|\,\mathscr{M}_B(x)\,|^2}$$

Matrix element ratio at each m, Γ can be obtained from Madgraph

Toy example: Inference at parton level

$$r_{\mathrm{ME}}(x|\theta) = \frac{|\mathscr{M}_S(x|\theta)|^2}{|\mathscr{M}_B(x)|^2}$$

Matrix element ratio at each m, Γ can be obtained from Madgraph

$$\log \Lambda_{\mathrm{shape}}(g,\theta) = \sum_{i=1}^{n} \log\left[1 + g^8 r_{\mathrm{ME}}(x_i|\theta)\right] - n\log\left[1 + \frac{g^8\sigma_S(\theta)}{\sigma_B}\right]$$

Exact log likelihood ratio from Madgraph

Toy example: Inference at parton level

$$r_{\rm ME}(x\,|\,\theta) = \frac{|\,\mathcal{M}_S(x\,|\,\theta)\,|^2}{|\,\mathcal{M}_B(x)\,|^2}$$

Matrix element ratio at each m, Γ can be obtained from Madgraph

Compare:

$$\log \Lambda_{\rm shape}(g,\theta) = \sum_{i=1}^{n} \log\left[1 + g^8 r_{\rm ME}(x_i\,|\,\theta)\right] - n\log\left[1 + \frac{g^8\sigma_S(\theta)}{\sigma_B}\right]$$

Exact log likelihood ratio from Madgraph

$$\log \Lambda_{\rm shape}(g,\theta) = \sum_{i=1}^{n} \log\left[1 + \frac{g^8\sigma_S(\theta)}{\sigma_B} r(x_i\,|\,\theta)\right] - n\log\left[1 + \frac{g^8\sigma_S(\theta)}{\sigma_B}\right]$$

Classifier based log-likelihood ratio

Toy example: Inference at parton level

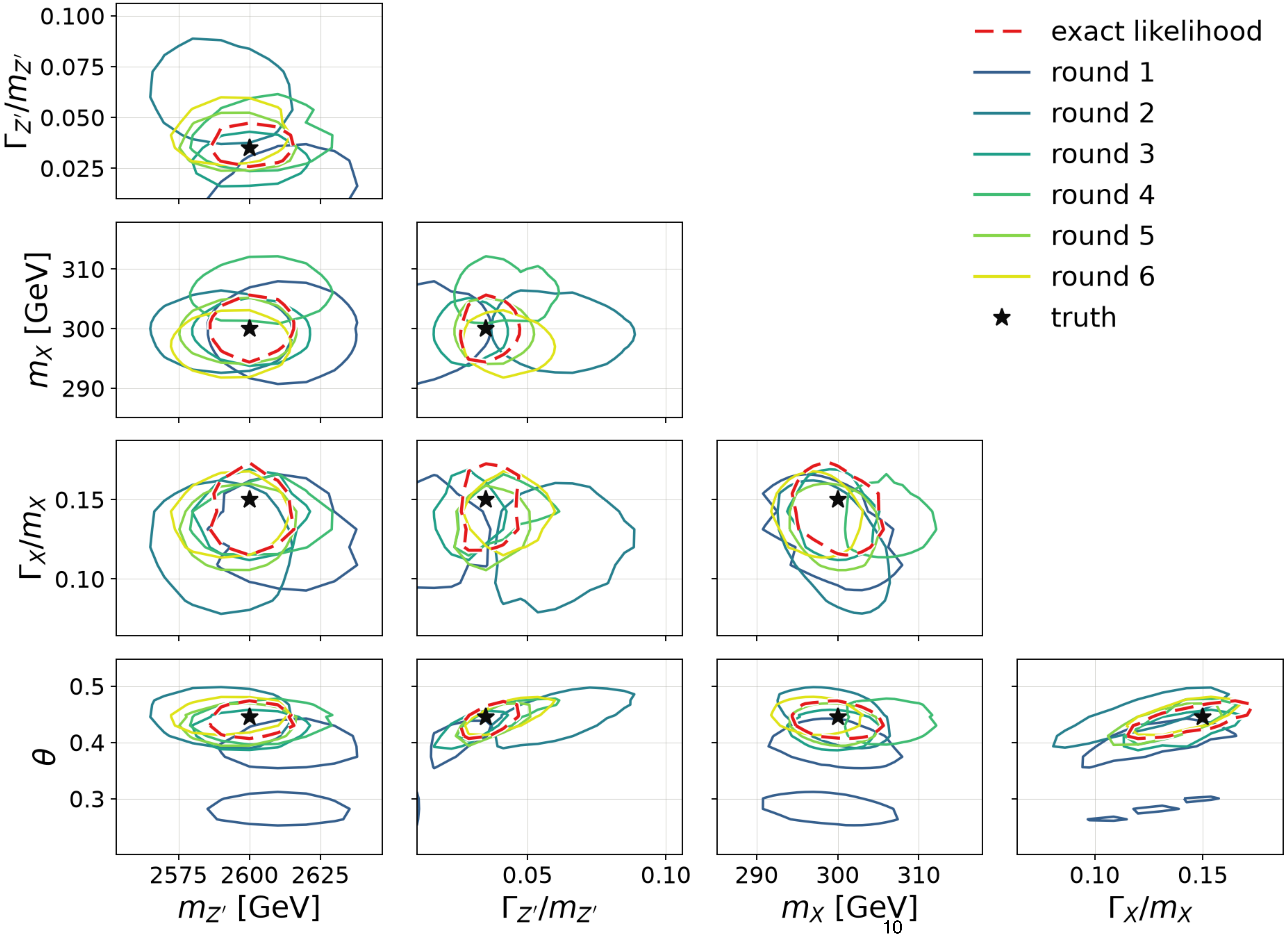
Signal: $u\bar{u} \to Z' \to X\bar{X} \to (u\bar{d})(\bar{u}d)$

Background: $u\bar{u} \to (u\bar{d})(\bar{u}d)$

$\theta: \quad \theta_{shape} = \left(m_{Z'}, \Gamma_{Z'}, m_X, \Gamma_X\right)$ and g

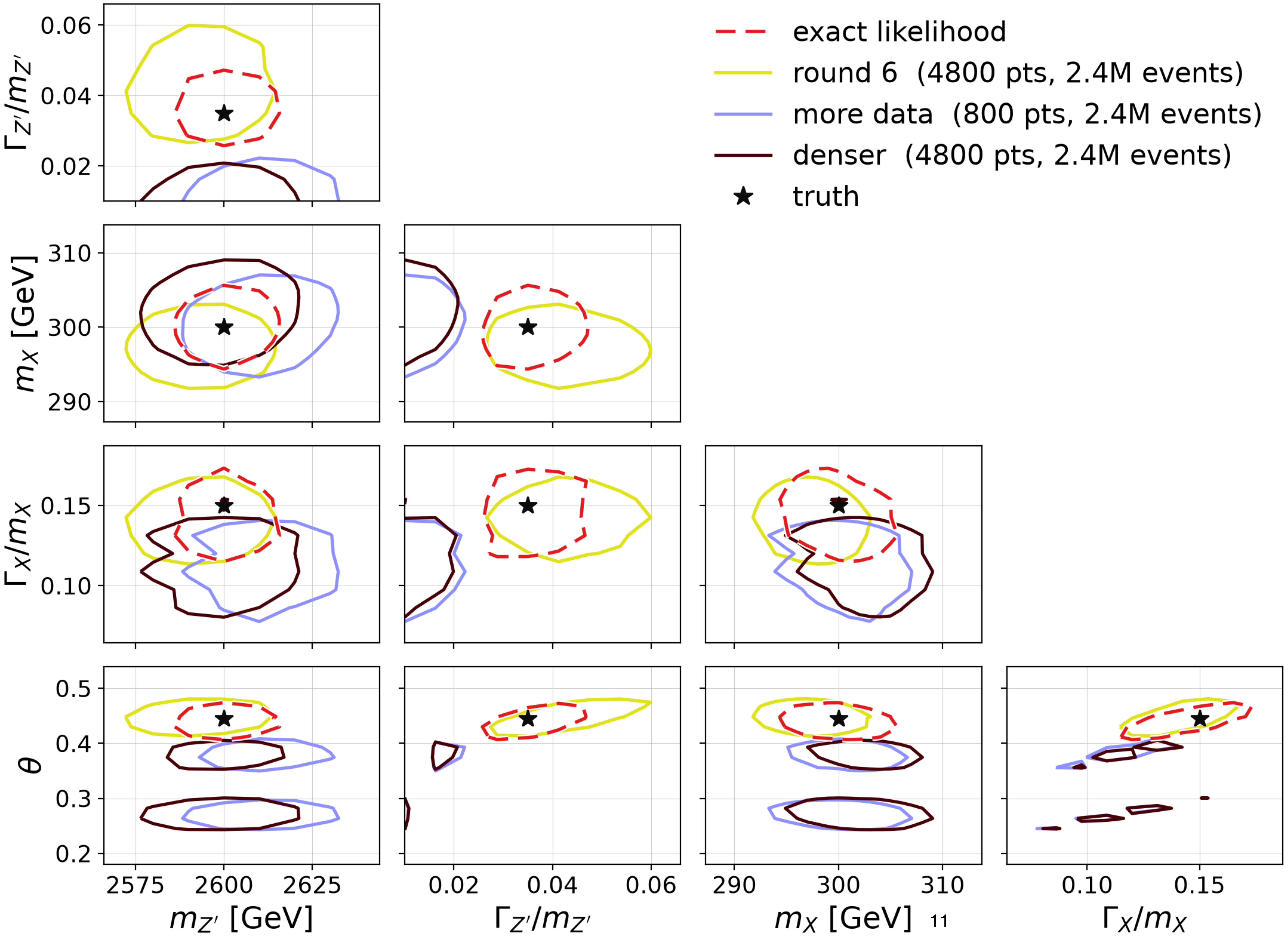
$m_{Z'} = 2600$ GeV, $\dfrac{\Gamma_{Z'}}{m_{Z'}} = 0.035,$ $\quad m_X = 300$ GeV, $\dfrac{\Gamma_X}{m_X} = 0.15$

Toy example: Inference at parton level



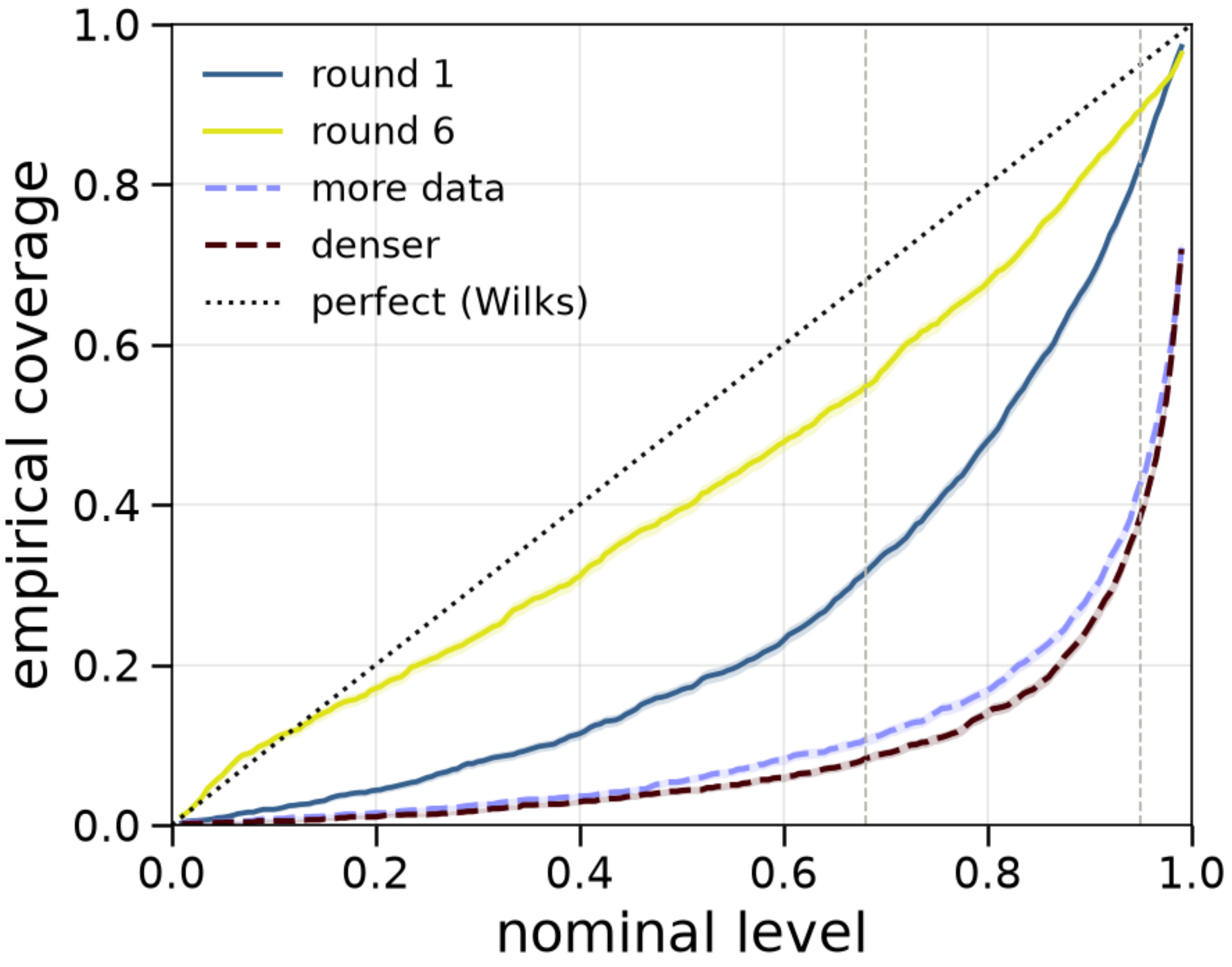
Rounds converge to true confidence regions!

Toy example: Inference at parton level



Iterative rounds do better single round with the same simulation budget!

Toy example: Inference at parton level



Coverage improves with rounds!

Inference at reco-level

Signal: $pp \to Y \to HH \to (t\bar{t})(t\bar{t})$

Background: $pp \to$ QCD Di-jet

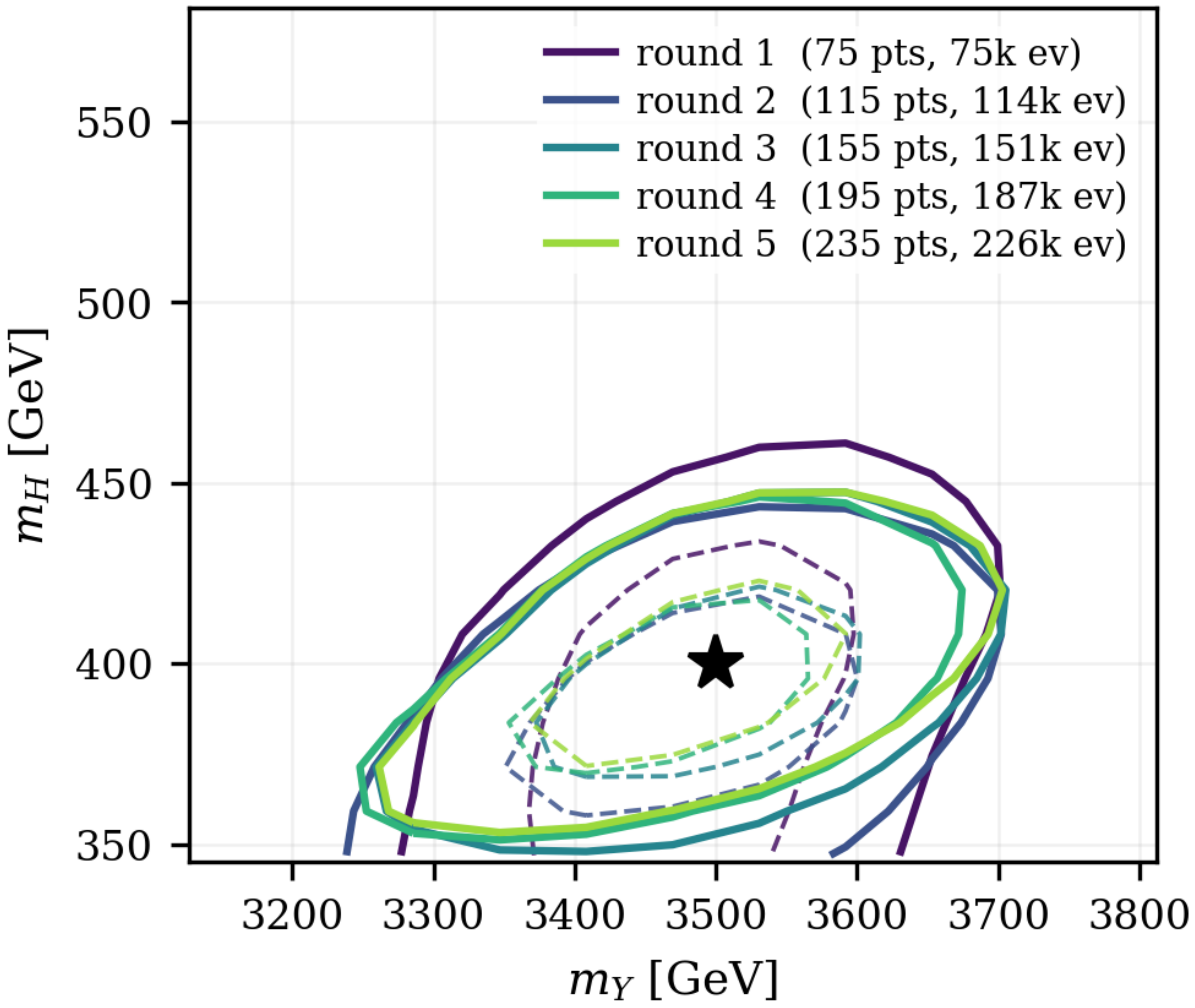
$\theta: \ \theta_{shape} = \left(m_Y, m_H\right)$ and σ_{sig}

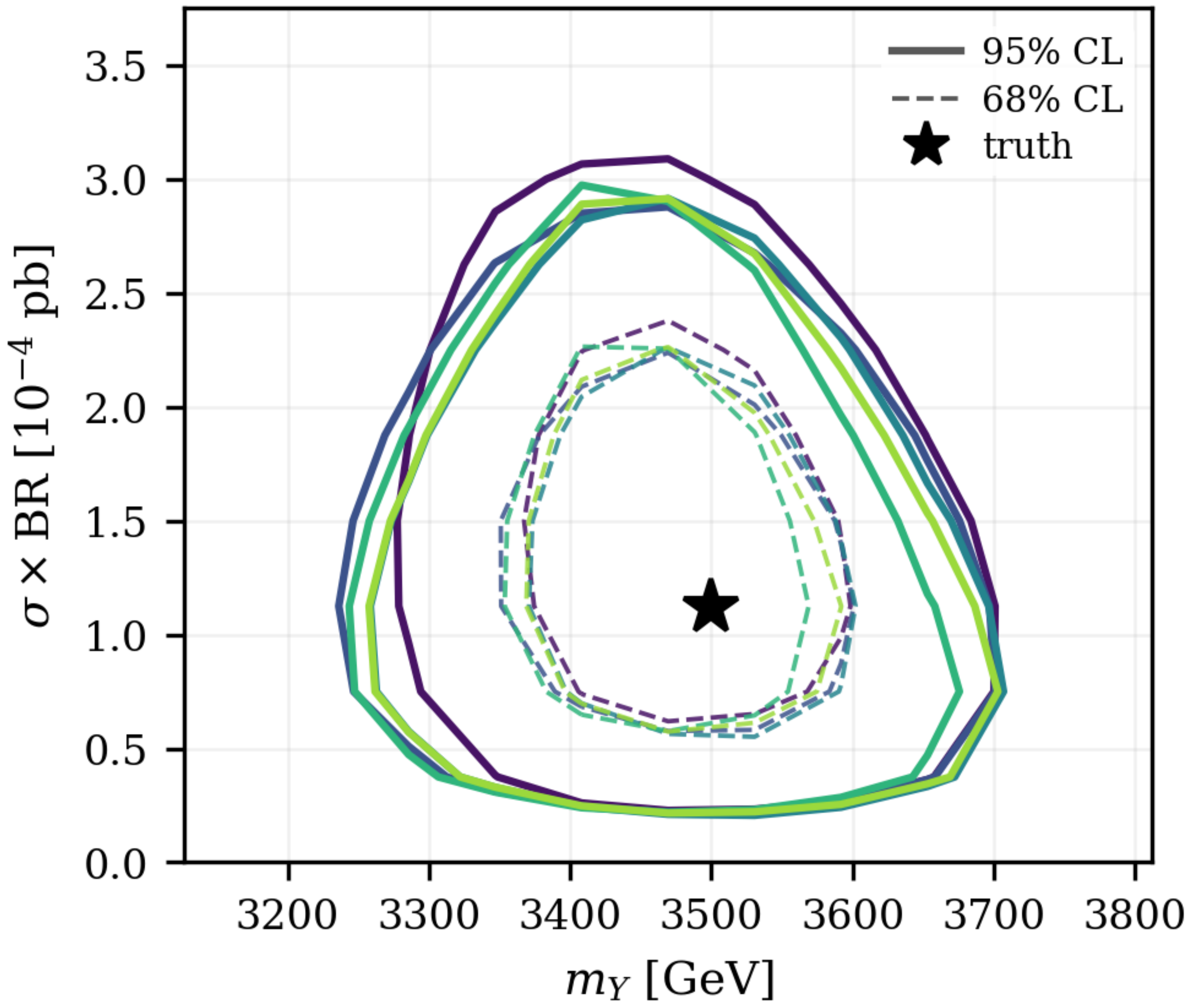
$m_Y = 3500$ GeV, $m_H = 400$ GeV $\sigma_{sig} = 0.010928$ pb

Parameterized classifier trained on full reconstructed jets

L-GATr-slim (arXiv:2512.17011)

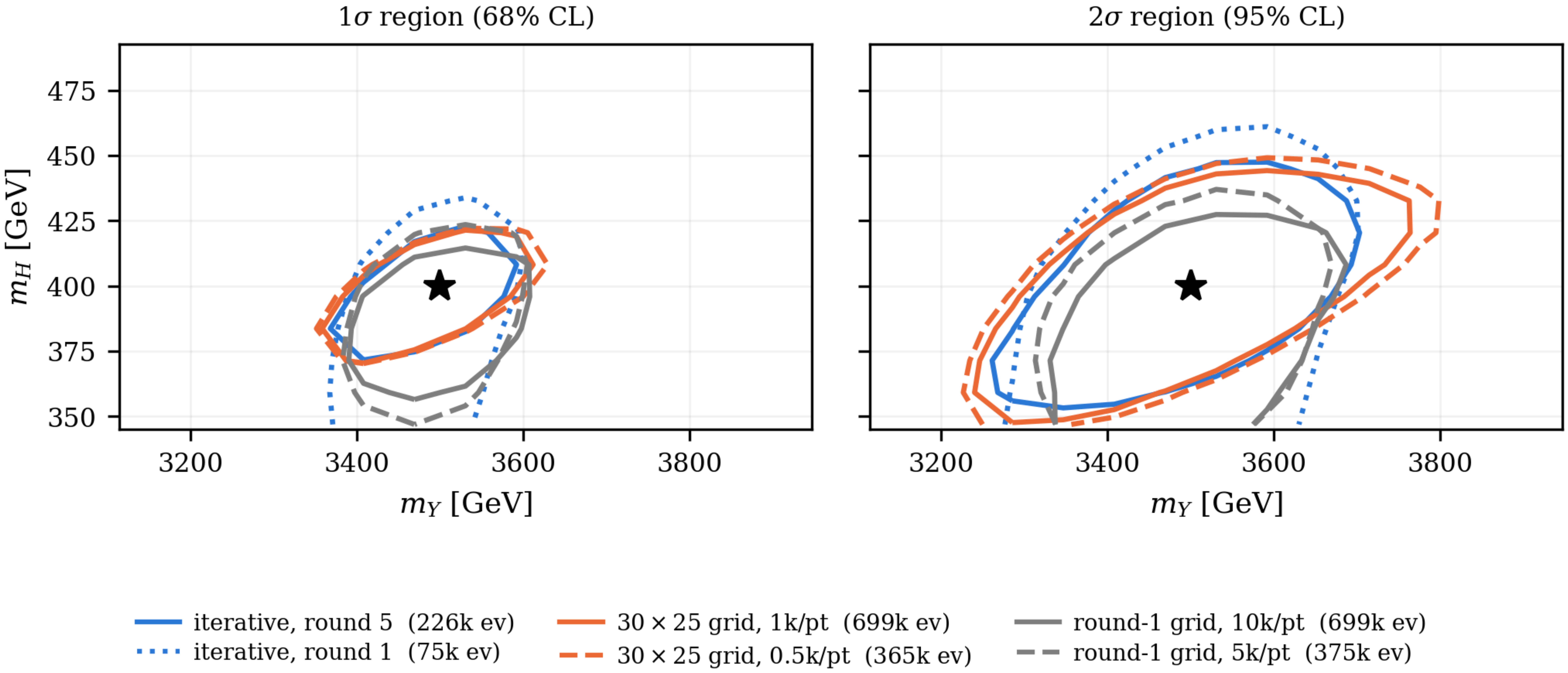
Inference at reco-level (Preliminary Results)





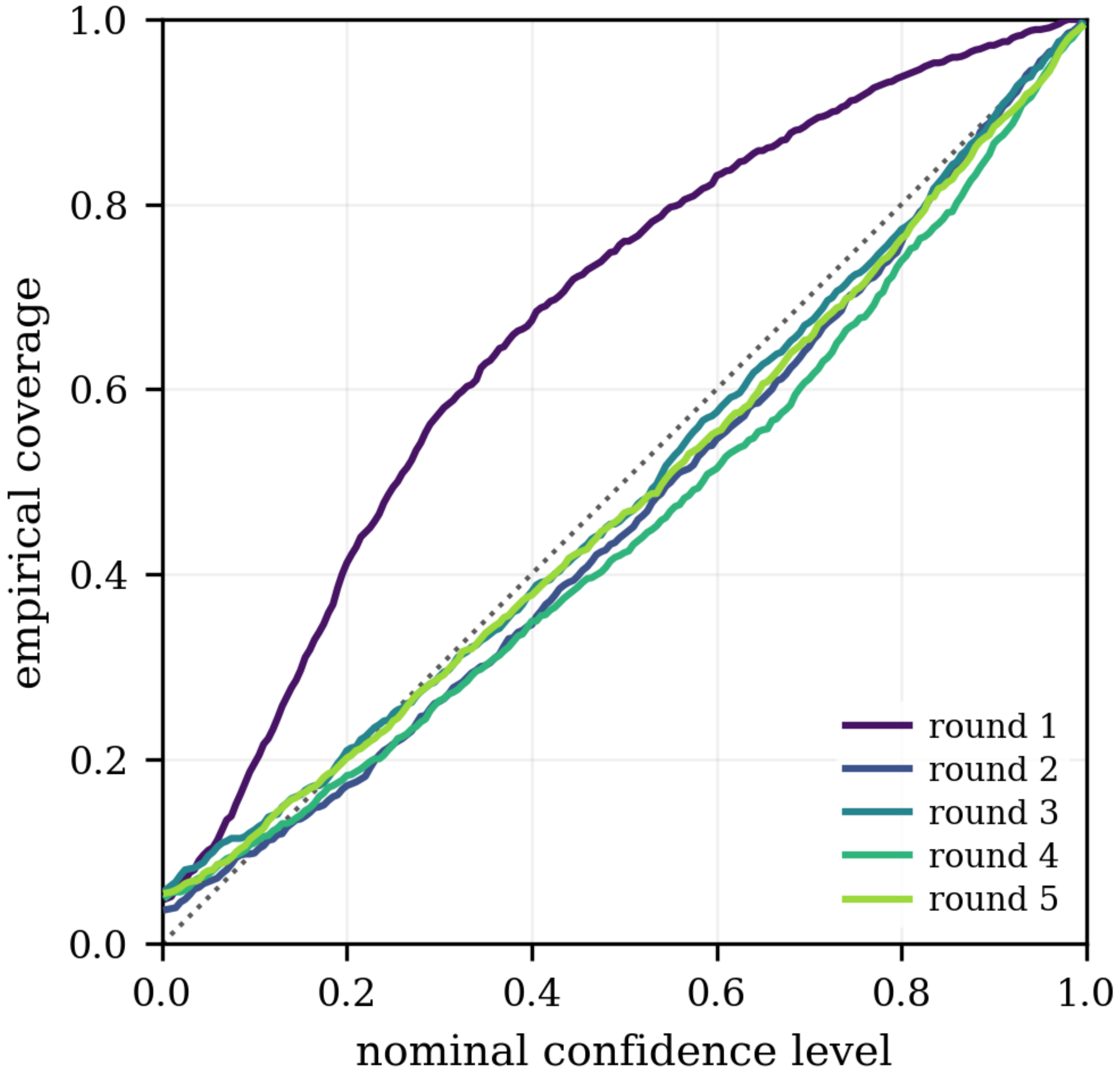
Rounds converge to tighter 68% confidence regions

Inference at reco-level (Preliminary Results)



Tighter 68% and 95% confidence regions than single round with the more simulations

Inference at reco-level (Preliminary Results)



Iterations don't make the coverage under-confident or over-confident.

Conclusions

- For inferring mass/widths generating simulations can be a bottleneck.
- Sequential-SBI generates simulations for parameters closer to high likelihood regions, resulting in more efficient simulations
- For parton level and reco-level, iterative SBI results in better confidence regions as well as improved coverage

THANK YOU