Why so negative? A General Approach to Quasi-Probabilistic Likelihood Ratio Estimation with Negative Weighted Data

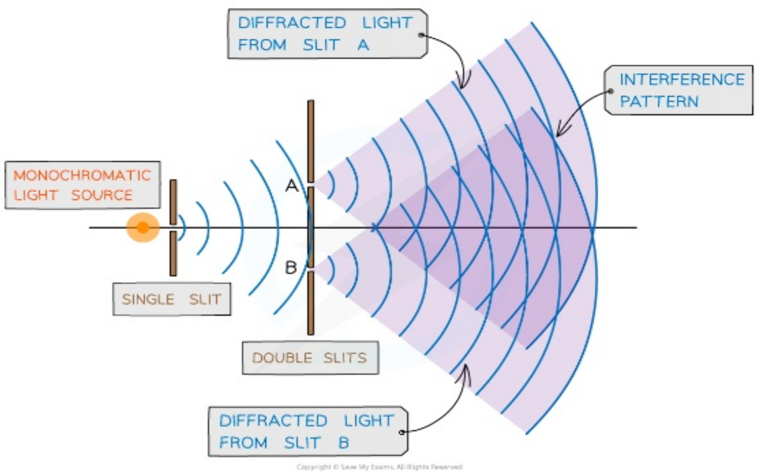
13-06-24

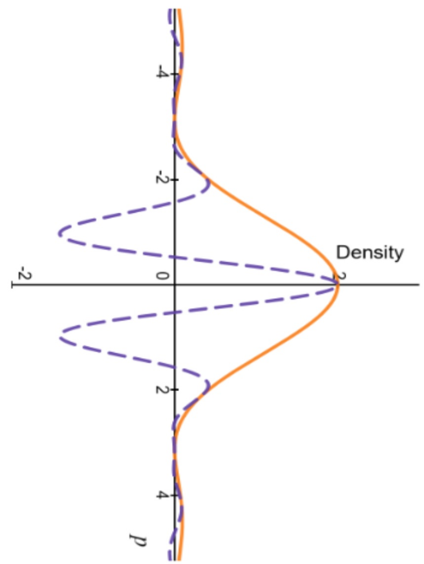
S.Jiggins (DESY)
M.Drnevich, Judith Katzy, Kyle Cranmer

Quasi-Probabilistic Distributions

→ **Quantum mechanical** systems are unique, in that there are objects that can be interpreted via quasi-probabilities that have ‘probabilistic-like terms’ that can be negative :

$$|\Phi_1(x) + \Phi_2(x)| = |\Phi_1(x)|^2 + |\Phi_2(x)|^2 + 2\mathcal{R}\left(\Phi_1(x)\Phi^*_2(x)\right)$$





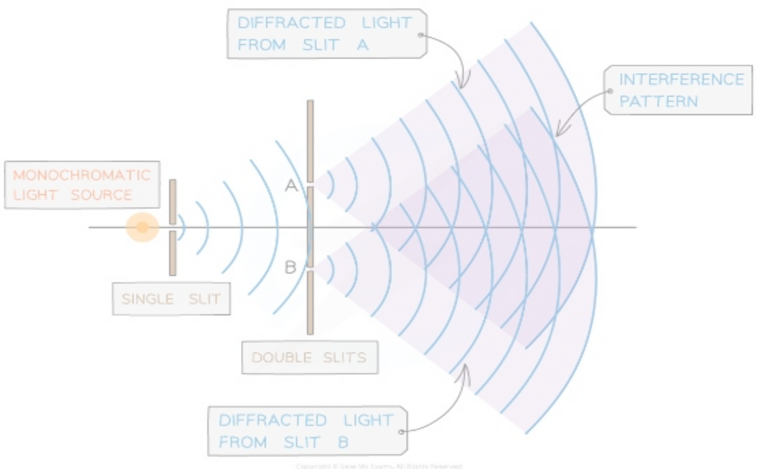
“It is usual to suppose that, since the probabilities of events must be positive, a theory which gives negative numbers for such quantities must be absurd … By discussing a number of examples, I hope to show that they are entirely rational of course, and that their use simplifies calculation and thought in a number of applications.”

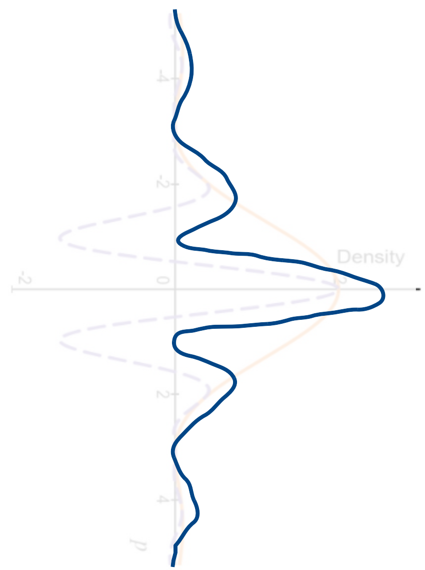
- Richard Feynman, Negative Probability https://cds.cern.ch/record/154856

Quasi-Probabilistic Distributions

→ **Quantum mechanical** systems are unique, in that there are objects that can be interpreted via quasi-probabilities that have 'probabilistic-like terms' that can be negative :

$$|\Phi_1(x) + \Phi_2(x)| = |\Phi_1(x)|^2 + |\Phi_2(x)|^2 + 2\mathcal{R}(\Phi_1(x)\Phi^*{}_2(x))$$





→ **Quantum mechanical** observations of an observable (G) are nothing more than averages of all states that contribute:

$$\langle\psi|\hat{G}|\psi\rangle = \mathrm{Tr}(\hat{\rho}\hat{G}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{W(x,p)}\, g(x,p)\,dx\,dp$$

Quasi-probability density (Wigner) function:
$-\infty < W(x,p) < \infty$

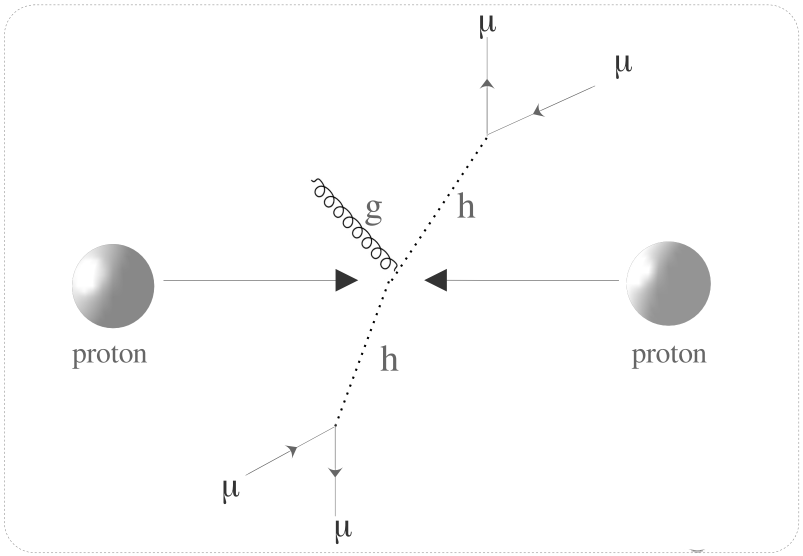
Remark!

The probability of an observation i of observable G, $p(G_i)$, will always be positive.

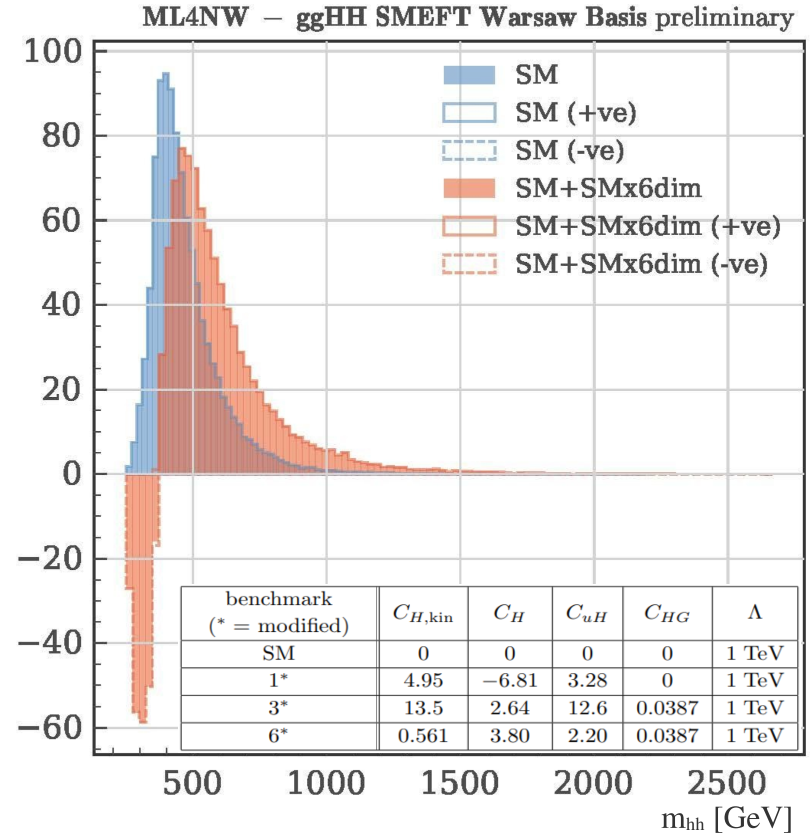
→ **Gleason's Theorem**

Quasi-Probabilistic Distributions

→ **Quantum mechanical** systems are unique, in that there are objects that can be interpreted via quasi-probabilities that have 'probabilistic-like terms' that can be negative :



$$\mathcal{M} = \mathcal{M}_{\rm SM} + \mathcal{M}_{\rm dim6} + \mathcal{M}_{\rm dim6^2} ,$$



benchmark (* = modified)	$C_{H,\rm kin}$	C_H	C_{uH}	C_{HG}	Λ
SM	0	0	0	0	1 TeV
1*	4.95	−6.81	3.28	0	1 TeV
3*	13.5	2.64	12.6	0.0387	1 TeV
6*	0.561	3.80	2.20	0.0387	1 TeV

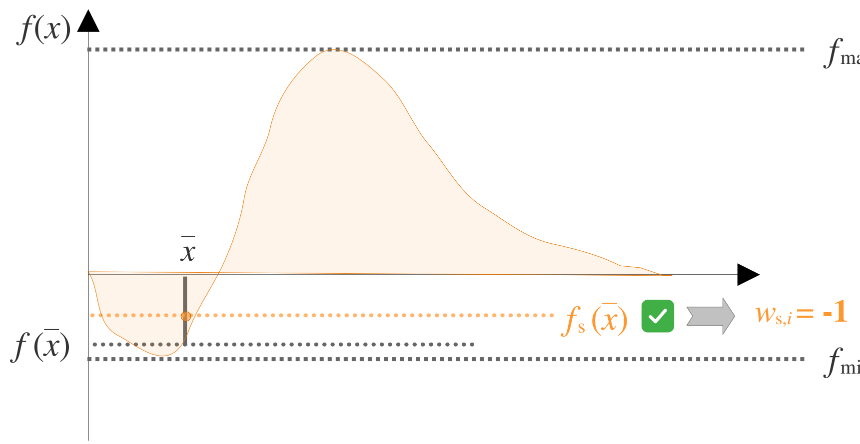
→ **Quantum mechanical** observations of an observable (G) are nothing more than averages of all states that contribute:

$$\langle\psi|\hat{G}|\psi\rangle = \mathrm{Tr}(\hat{\rho}\hat{G}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{W(x,p)}_{} g(x,p)\,dx\,dp$$

Quasi-probability density (Wigner) function:
$-\infty < W(x,p) < \infty$

Quasi-Probabilities: The negative weight problem

Synthetic Data Generation: Monte Carlo



(1) Uniformly sample points, $\bar{x} = (x_1, \ldots, x_i)$, uniformly on $[a,b] \otimes [f_{min}, f_{max}]$:

$f_s(\bar{x}) \in [f_{min}, f_{max}]$

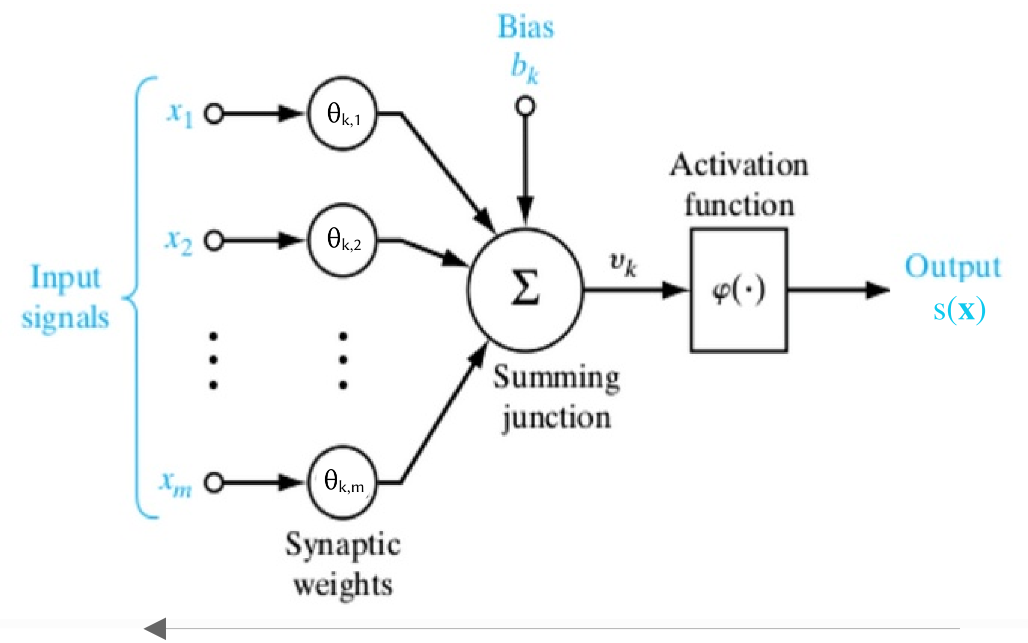
(2) Accept event based on:
$f_s(\bar{x}) < |f(\bar{x})|$

(3) Assign a weight to the event:
$w_{s,i} = \pm 1$

(4) Repeat

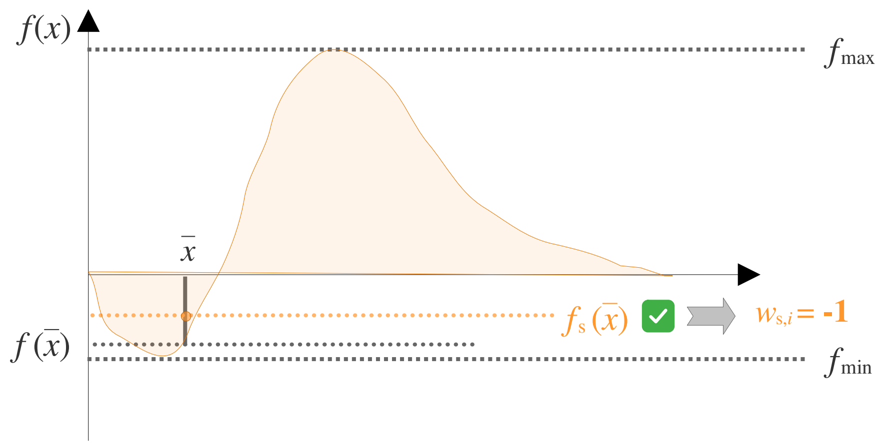
Quasi-Probabilities: The negative weight problem

Parameter updates of a neural network use the weighted loss during backwards propagation:



$$\theta^{k+1} = \theta^k - \gamma \sum^{N_i} w_i \cdot \nabla_\theta \mathcal{L}(s(\mathbf{x}_i;\theta), y_i)$$

Synthetic Data Generation: Monte Carlo



(1) Uniformly sample points, $\bar{x} = (x_1, \ldots, x_i)$, uniformly on $[a,b] \otimes [f_{\min}, f_{\max}]$:

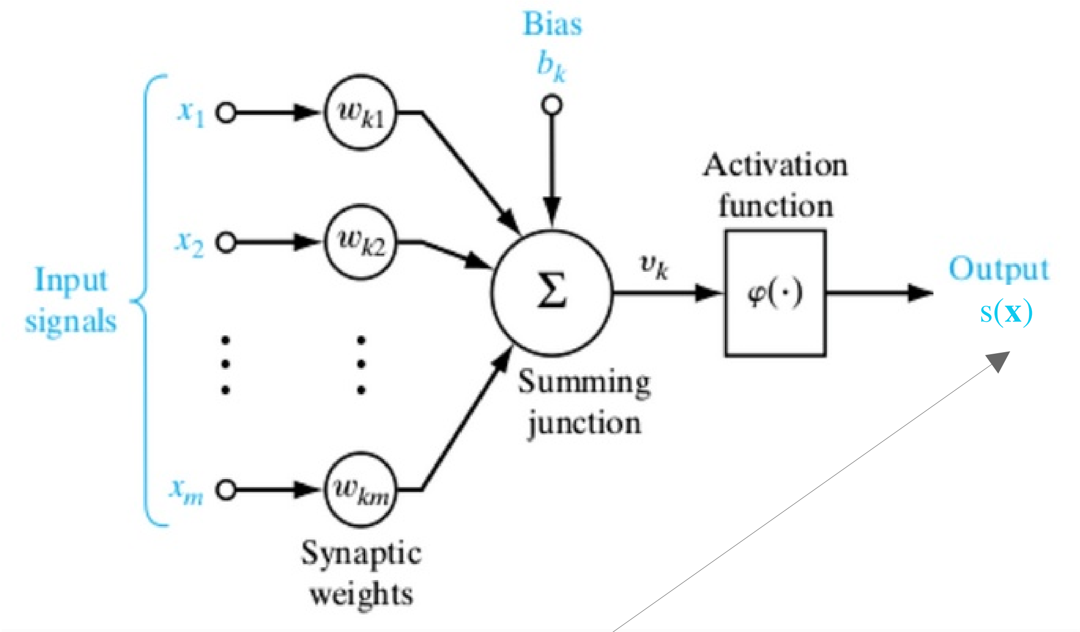
$f_s(\bar{x}) \in [f_{\min}, f_{\max}]$

(2) Accept event based on:
$f_s(\bar{x}) < |f(\bar{x})|$

(3) Assign a weight to the event:
$w_{s,i} = \pm 1$

(4) Repeat

A Fundamental Conflict of Statistics, Probability, and Information



→ Output of the neural based probabilistic models are concerned only with the s(x) ∈ (0, ∞)

Statistics & Probability Theory

In probability theory the probability triplet (Ω, F, P) adheres to 3 Kolmogorov[1] axioms with the most relevant being axiom 1:

$$P[X \in A] = \int_A p_X \, d^n x$$

$$p_X \geq 0 \ \forall x$$

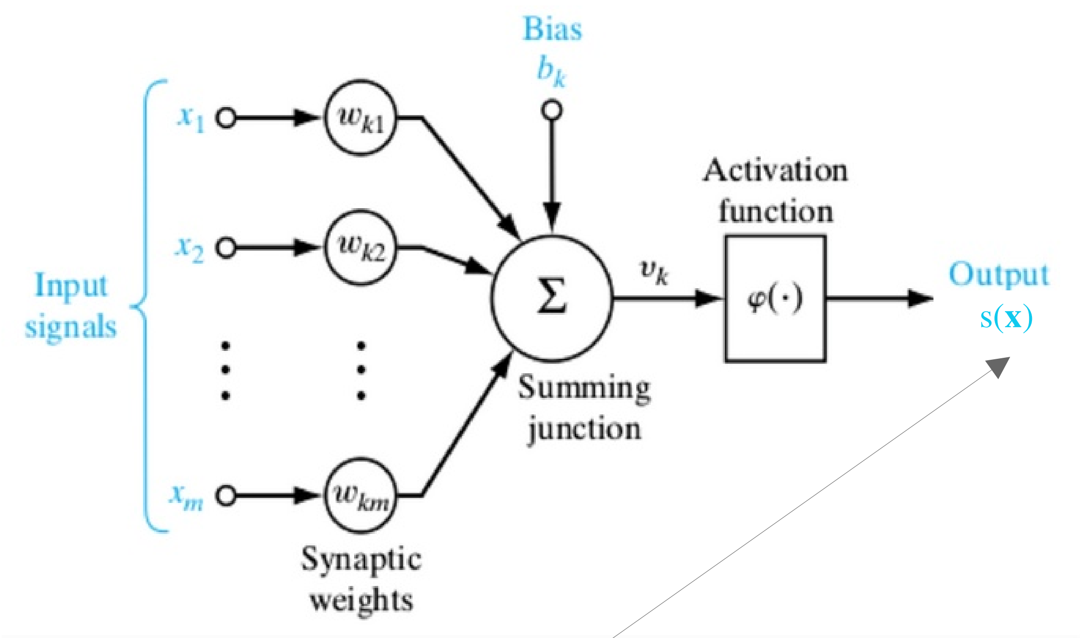
Information & Measure Theory

Entropy-based measures:
Average level of *information/surprise* inherent to a random variables outcome *(in F*-space*):*

$$H(X) := -\int p_X(x) \log(p_X(x)) \, dx$$

[1] A. Kolmogorov, Grundbegriffe der wahrscheinlichkeitsrechnung, 1 (Springer Berlin, Heidelberg, 1933), pp. V, 62.
[2] D. S. Modha and Y. Fainman, "A learning law for density estimation," in IEEE Transactions on Neural Networks, vol. 5, no. 3, pp. 519-523, May 1994, doi: 10.1109/72.286931

A Fundamental Conflict of Statistics, Probability, and Information



→ Output of the neural based probabilistic models must allow for s(x) ∈ (-∞, ∞)

Statistics & Probability Theory

In probability theory the probability triplet (Ω, F, P) adheres to 3 Kolmogorov[1] axioms with the most relevant being axiom 1:

$$P[X \in A] = \int_A p_X d^n x$$

$$p_X \in (-\infty, \infty)$$

Information & Measure Theory

Entropy-based measures:
Average level of *information/surprise* inherent to a random variables outcome *(in F*-space*):*

$$H(X) := -\int p_X(x) \log(p_X(x))\, dx$$

[1] A. Kolmogorov, Grundbegriffe der wahrscheinlichkeitsrechnung, 1 (Springer Berlin, Heidelberg, 1933), pp. V, 62.
[2] D. S. Modha and Y. Fainman, "A learning law for density estimation," in IEEE Transactions on Neural Networks, vol. 5, no. 3, pp. 519-523, May 1994, doi: 10.1109/72.286931

Context: Likelihood Function

→ Likelihood function is key to the *scientific method:*

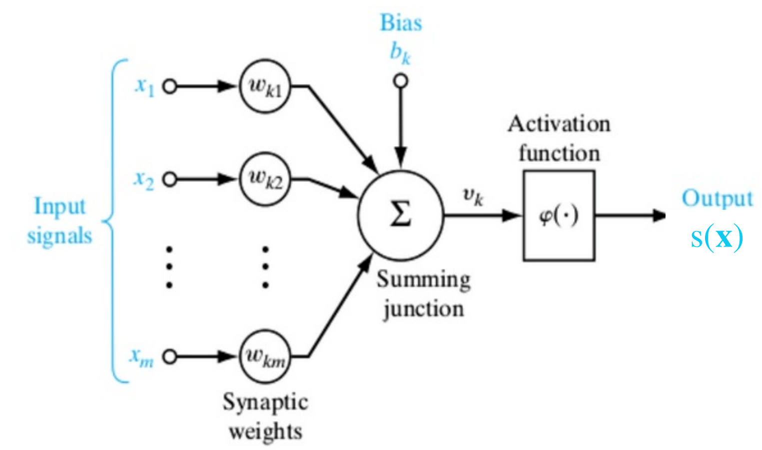
$$\mathcal{L}(\theta|x) = p_\theta(x) = P(X = x|\theta)$$

→ This is due to its prolific use in *statistical inferencing* problems via its ratio form:

$$r(x|\theta_1, \theta_2) = \frac{\mathcal{L}(\theta_1|x)}{\mathcal{L}(\theta_0|x)}$$

Neural Likelihood Ratio Estimation

'Ratio Trick'



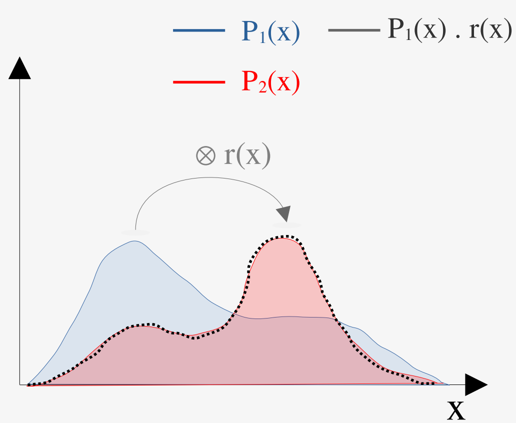
For a neural network with configurable set $\Phi = \{w_i\}^N_{i=1}$ parameters and a given loss functional L(s) of the form:

$$\mathcal{L}(s) = -\int_{\mathbb{R}^n} d^n x \left(p(x|\theta_0) \cdot \ln(s(x)) + p(x|\theta_1) \cdot \ln(1 - s(x))\right)$$

The extrema (δL(s)/δs =0) of this general loss yields:

$$\frac{1 - s(x)}{s(x)} = \frac{p(x|\theta_1)}{p(x|\theta_0)} = r(x|\theta_0, \theta_1)$$

Domain Adaptation



Use the density ratio r(x) as a mapping function between distribution:

r(x): P1(x) → P2(x)

Quasi-Probabilities & ML

Signed Probability Measures

→ **Decompose probability measure** into a signed measure:

$$\mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n \xrightarrow[\mathbf{P} = \mathbf{P}_+ - \mathbf{P}_-]{} \mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n = \int_A (p_+(\mathbf{x}) - p_-(\mathbf{x})) d\mathbf{x}^n$$

Known as the *Jordan Decomposition.*

Signed Probability Measures

→ **Decompose probability measure** into a signed measure:

$$\mathbf{P}[X \in A] = \int_A q_X(\mathbf{x})d\mathbf{x}^n \quad \xrightarrow[\mathbf{P}=\mathbf{P}_+ - \mathbf{P}_-]{} \quad \mathbf{P}[X \in A] = \int_A q_X(\mathbf{x})d\mathbf{x}^n = \int_A (p_+(\mathbf{x}) - p_-(\mathbf{x}))d\mathbf{x}^n$$

Known as the *Jordan Decomposition.*

→ **Signed Mixture Model** decomposition translates to a *mixture* of likelihood ratios:

$$r(x|\theta_0,\theta_1,\mathbf{c}) = \frac{\sum_i^{[+,-]} c_{i,1}\, p_i(\boldsymbol{x}|\theta_1)}{\sum_j^{[+,-]} c_{j,0}\, p_j(\boldsymbol{x}|\theta_0)} \quad \xrightarrow[\text{classes}]{\text{For 2}}$$

$$r(x;\mathbf{c}) = \left[\frac{c_0 p_+(x|Y=0)}{c_1 p_+(x|Y=1)} + \frac{(1-c_0)p_-(x|Y=0)}{c_1 p_+(x|Y=1)}\right]^{-1} + \left[\frac{c_0 p_+(x|Y=0)}{(1-c_1)p_-(x|Y=1)} + \frac{(1-c_0)p_-(x|Y=0)}{(1-c_1)p_-(x|Y=1)}\right]^{-1}$$

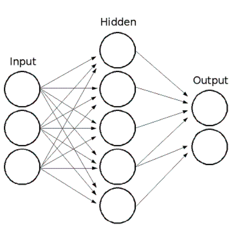
$$r(x|y_0,y_1,c) = \sum_i \left[\sum_j \frac{c_{j,0}}{c_{i,1}} \cdot r_{i,j}\right]^{-1}$$

Co-efficients defined by the normalised ratio of +ve/-ve subsets of the data to the total class weight

$$c_i \sim \frac{\sum w_y^{\pm}}{\sum w_y}$$

Train separate & unique calibrated NLREs (see CARL) for various permutations of:

$$\frac{P_{[+,-]}(y=0\,|x)}{P_{[+,-]}(y=1\,|x)} =$$



Signed Probability Measures

→ **Decompose probability measure** into a signed measure:

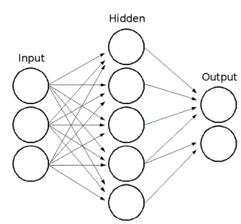
$$\mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n \xrightarrow[\mathbf{P}=\mathbf{P}_+ - \mathbf{P}_-]{} \mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n = \int_A (p_+(\mathbf{x}) - p_-(\mathbf{x})) d\mathbf{x}^n$$

Known as the *Jordan Decomposition.*

→ **Signed Mixture Model** decomposition translates to a *mixture* of likelihood ratios:

$$r(x|\theta_0, \theta_1, \mathbf{c}) = \frac{\sum_i^{[+,-]} c_{i,1}\, p_i(\boldsymbol{x}|\theta_1)}{\sum_j^{[+,-]} c_{j,0}\, p_j(\boldsymbol{x}|\theta_0)}$$

$$r(x|y_0, y_1, c) = \sum_i \left[\sum_j \underbrace{\frac{c_{j,0}}{c_{i,1}}}\ \cdots \right]^{-1}$$



Co-efficients defined by the normalised ratio of +ve/-ve subsets of the data to the total class weight

$$c_i \sim \frac{\sum w_y^{\pm}}{\sum w_y}$$

Train separate & unique calibrated NLREs (see CARL) for various permutations of:

$$r_{i,j} = \frac{P_{[+,-]}(y=0\,|x)}{P_{[+,-]}(y=1\,|x)}$$

Two Key Points:

(1)
The sub-likelihood ratios are translated to the positive domain by setting the weights of all data to the absolute value:
$w_i \longrightarrow |w_i|$

Signed Probability Measures

→ **Decompose probability measure** into a signed measure:

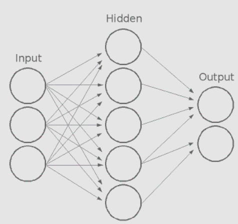
$$\mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n \quad \xrightarrow[\mathbf{P} = \mathbf{P}_+ - \mathbf{P}_-]{} \quad \mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n = \int_A (p_+(\mathbf{x}) - p_-(\mathbf{x})) d\mathbf{x}^n$$

Known as the *Jordan Decomposition.*

→ **Signed Mixture Model** decomposition translates to a *mixture* of likelihood ratios:

$$r(x|\theta_0, \theta_1, \mathbf{c}) = \frac{\sum_i^{[+,-]} c_{i,1}\, p_i(\boldsymbol{x}|\theta_1)}{\sum_j^{[+,-]} c_{j,0}\, p_j(\boldsymbol{x}|\theta_0)}$$

$$r(x|y_0, y_1, c) = \sum_i \left(\sum_j \underbrace{\frac{c_{j,0}}{c_{i,1}}}\ \underbrace{[\text{NN}]} \right)^{-1}$$



Co-efficients defined by the normalised ratio of +ve/-ve subsets of the data to the total class weight

$$c_i \sim \frac{\sum w_y^{\pm}}{\sum w_y}$$

Train separate & unique calibrated NLREs (see CARL) for various permutations of:

$$r_{i,j} = \frac{P_{[+,-]}(y=0\,|x)}{P_{[+,-]}(y=1\,|x)}$$

Two Key Points:

(1)
The sub-likelihood ratios are translated to the positive domain by setting the weights of all data to the absolute value:
$w_i \longrightarrow |w_i|$

(2)
All signed information about the prob. measure is contained in the constants

Signed Probability Measures

→ **Decompose probability measure** into a signed measure:

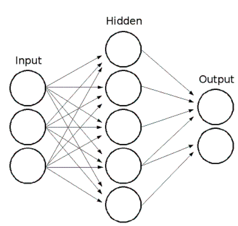
$$\mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n \quad \xrightarrow[\mathbf{P} = \mathbf{P}_+ - \mathbf{P}_-]{} \quad \mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n = \int_A (p_+(\mathbf{x}) - p_-(\mathbf{x})) d\mathbf{x}^n$$

Known as the *Jordan Decomposition.*

→ **Signed Mixture Model** decomposition translates to a *mixture* of likelihood ratios:

$$r(x|\theta_0, \theta_1, \mathbf{c}) = \frac{\sum_i^{[+,-]} c_{i,1}\, p_i(\boldsymbol{x}|\theta_1)}{\sum_j^{[+,-]} c_{j,0}\, p_j(\boldsymbol{x}|\theta_0)}$$

$$r_q(x|y_0, y_1, c) = \sum_i \Bigg(\sum_j \frac{c_{j,0}}{c_{i,1}}$$



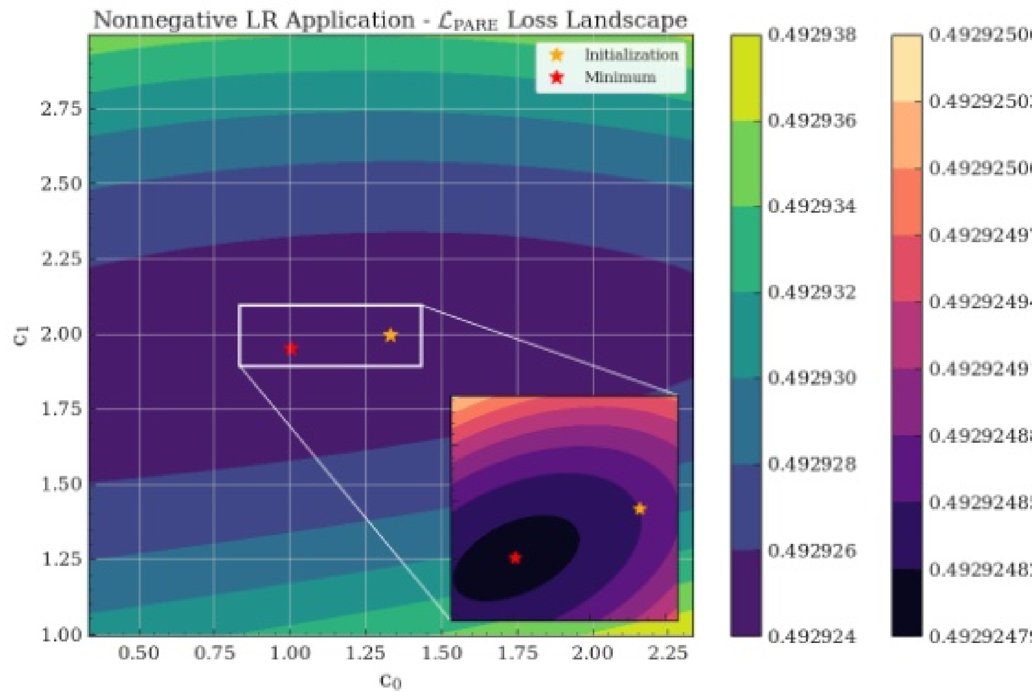
$$\Bigg)^{-1}$$

Co-efficients defined by the normalised ratio of +ve/-ve subsets of the data to the total class weight

$$c_i \sim \frac{\sum w_y^{\pm}}{\sum w_y}$$

Train separate & unique calibrated NLREs (see CARL) for various permutations of:

$$r_{i,j} = \frac{P^{[+,-]}(y=0\,|x)}{P^{[+,-]}(y=1\,|x)}$$



Optimise co-efficients and NNs using a new loss function, L_{PARE}:

$$\mathcal{L}_{PARE}(\hat{y}, y) \equiv (1 - \hat{y} \cdot y)^2$$

Transformation of the signed neural likelihood ratio estimator using class label $y_{0,1}$:

$$\hat{y}(\mathbf{x}) = \frac{y_0 + y_1 r_q(\mathbf{x})}{y_0^2 + y_1^2 r_q(\mathbf{x})}$$

How does this look?

Gaussian → Camel Function

Toy Model

Source: 2D Gaussian Distribution

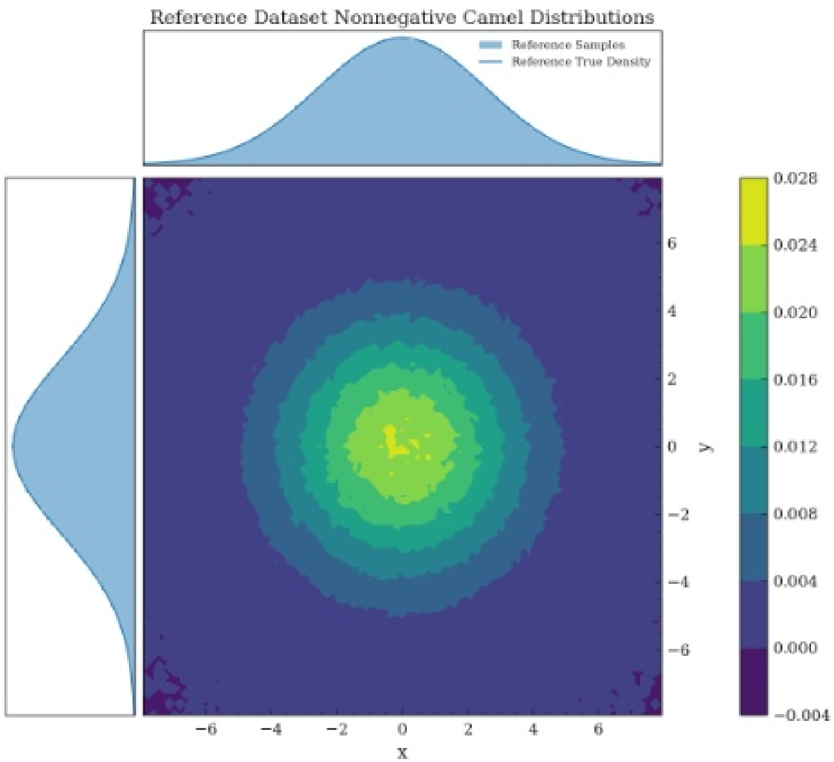
$$p(x,y;\sigma)=\frac{1}{2\pi\sigma^2}\cdot\exp\left(-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}\right)=p(x;\sigma)\,p(y;\sigma)$$

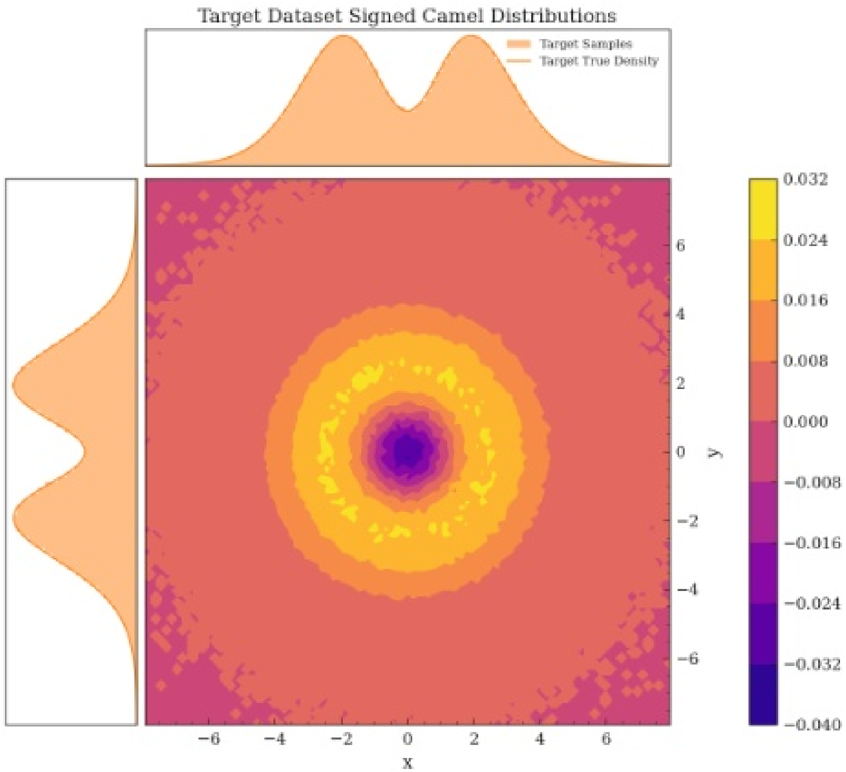
Where:
$\sigma = 2.5$

Target: 2D Camel Distribution

$$p(x,y;A,B,\sigma_1,\sigma_2)=\frac{1}{A+B}\cdot[A\cdot p(x,y;\sigma_1)+B\cdot p(x,y;\sigma_2)]$$

Where:
$\sigma_1 = 2$, $\sigma_2 = 1.2$
$A = 2$, $B = -1$





Toy Model

Source (y=0): 2D Gaussian Distribution

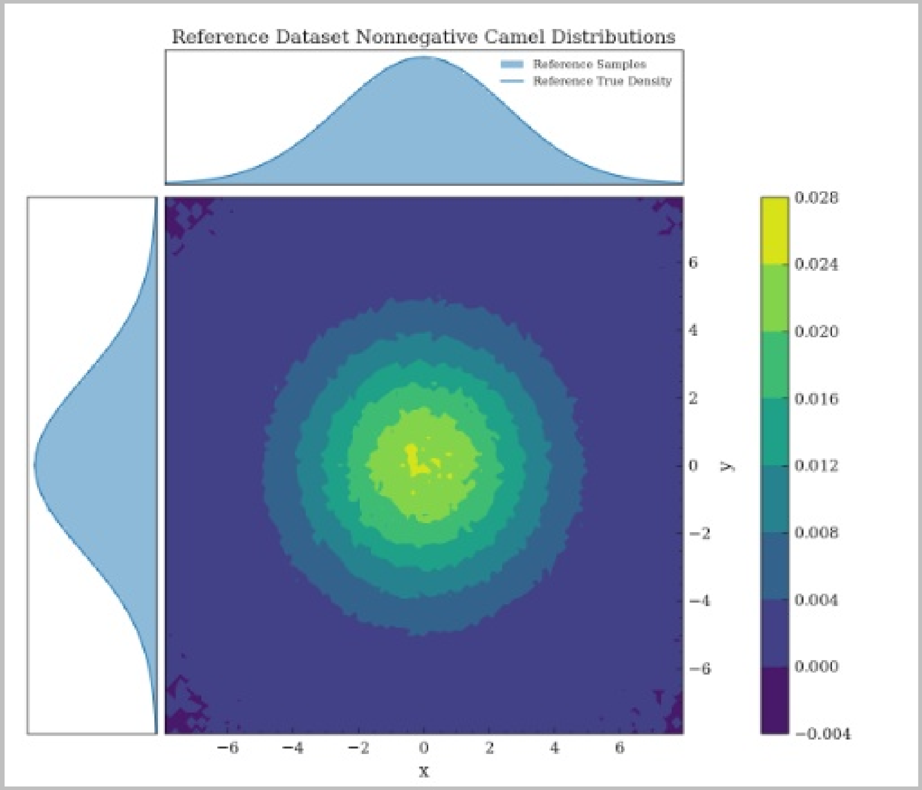
$$p(x,y;\sigma)=\frac{1}{2\pi\sigma^2}\cdot\exp\left(-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}\right)=p(x;\sigma)\,p(y;\sigma)$$

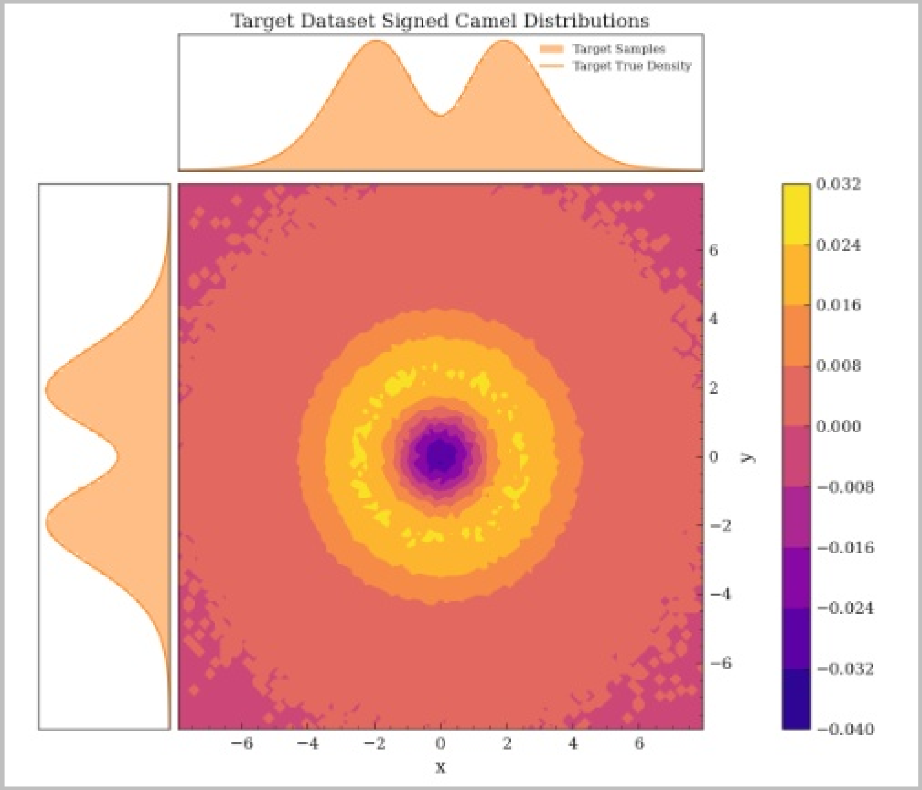
Where:
$\sigma = 2.5$

Target (y=1): 2D Camel Distribution

$$p(x,y;A,B,\sigma_1,\sigma_2)=\frac{1}{A+B}\cdot[A\cdot p(x,y;\sigma_1)+B\cdot p(x,y;\sigma_2)]$$

Where:
$\sigma_1 = 2$, $\sigma_2 = 1.2$
$A = 2$, $B = -1$





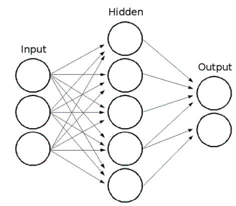
Map from y=0 → y=1 using NLRE:
$\hat{r}(x): P_0(x) \rightarrow P_1(x)$

Neural Models

1) Basic Model:

Standard MLP estimating the likelihood ratio:

$r(x|y_0, y_1) =$



$$\rightarrow \frac{s(x)}{1-s(x)}$$

2) Signed Mixture Model & Signed Mixture Model +:

4 standard MLPs comprising the signed mixture likelihood ratio with configurable co-efficients (c_i):

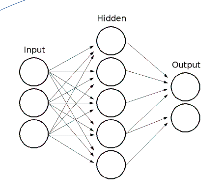
$$-\gamma \nabla_{c_i^t} \mathcal{L}(s(x; c_i^t)),$$

$\oplus$ backwards propagation update enabled for all 4 blocks

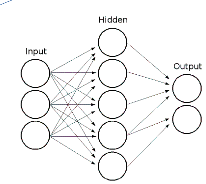
$$-\gamma \nabla_{\phi^t} \mathcal{L}(s(x; \phi^t))$$

$r(x|y_0, y_1, c) = \oplus$

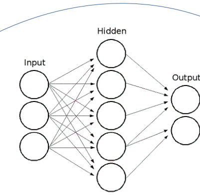
$\frac{c_0}{c_1}$.



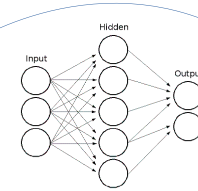
$\oplus$ $\frac{1-c_0}{c_1}$.



$\frac{c_0}{1-c_1}$.



$\oplus$ $\frac{1-c_0}{1-c_1}$.



3) Optimal:

Analytic solution for the optimal classifier/likelihood ratio:

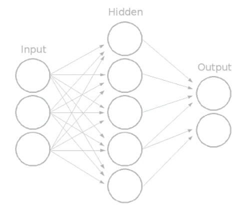
$$r(x|\theta_0, \theta_1, c) = \frac{p(x|\theta_1)}{p(x|\theta_0)}$$

Neural Models

1) Basic Model:

Standard MLP estimating the likelihood ratio:

$r(x|y_0, y_1) =$



$\rightarrow \dfrac{s(x)}{1-s(x)}$

Total Network Size:
~34.4 MB

2) Signed Mixture Model & Signed Mixture Model +:

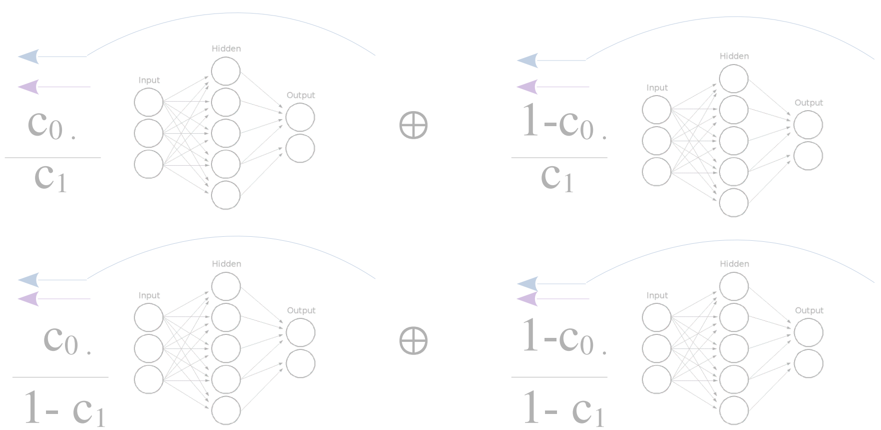
4 standard MLPs comprising the signed mixture likelihood ratio with configurable co-efficients (c_i):

$$-\gamma \nabla_{c_i^t} \mathcal{L}(s(x; c_i^t)),$$

$\oplus$ backwards propagation update enabled for all 4 blocks

$$-\gamma \nabla_{\phi^t} \mathcal{L}(s(x; \phi^t))$$

$r(x|y_0, y_1, c) = \oplus$



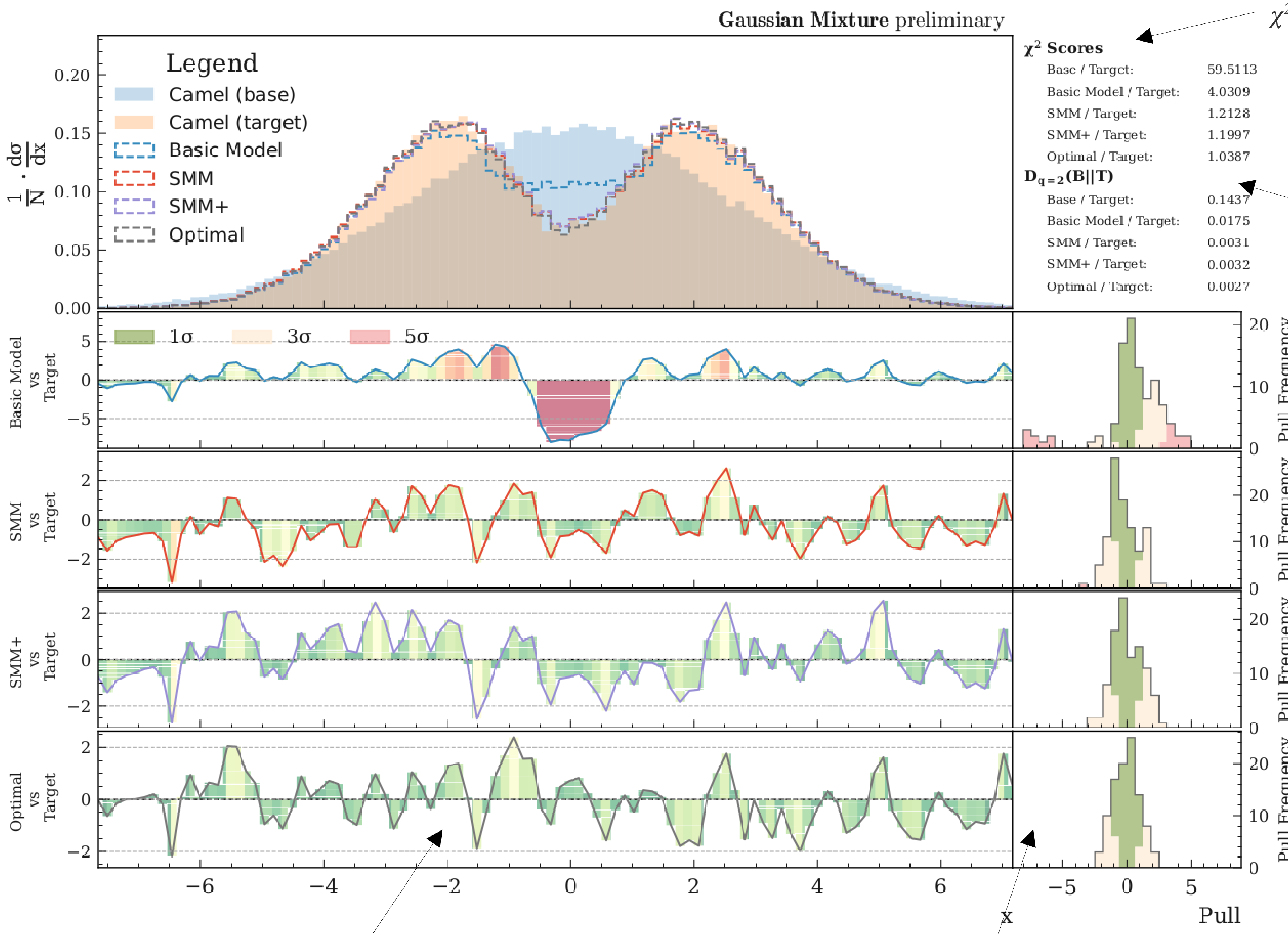
Total Network Size:
~35.9 MB

3) Optimal:

Analytic solution for the optimal classifier/likelihood ratio:

$$r(x|\theta_0, \theta_1, c) = \frac{p(x|\theta_1)}{p(x|\theta_0)}$$

Toy Model: Observation space



χ^2 Test Statistic

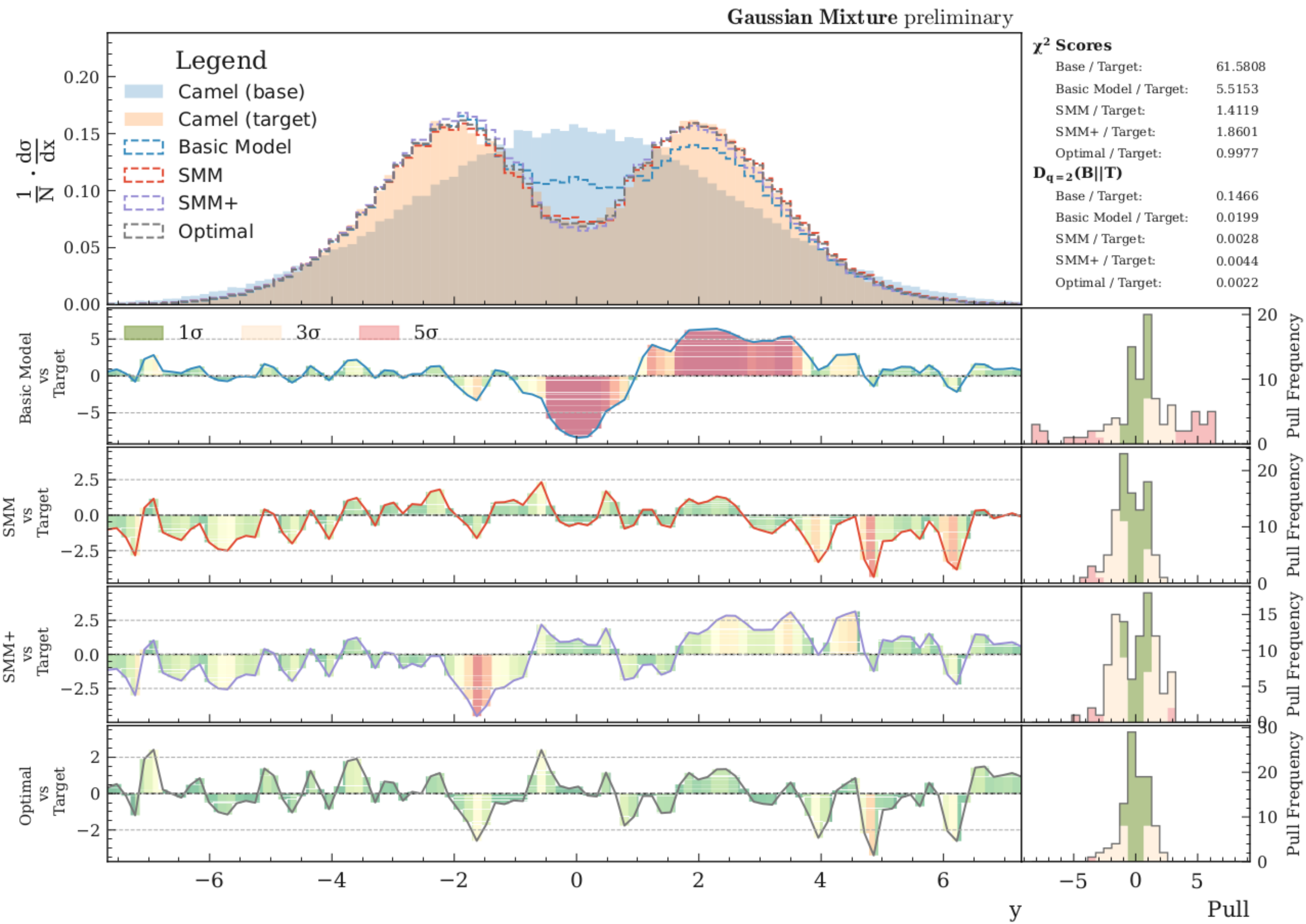
Q-deformed Relative Entropy:

$D_{q=2}(B||T)$

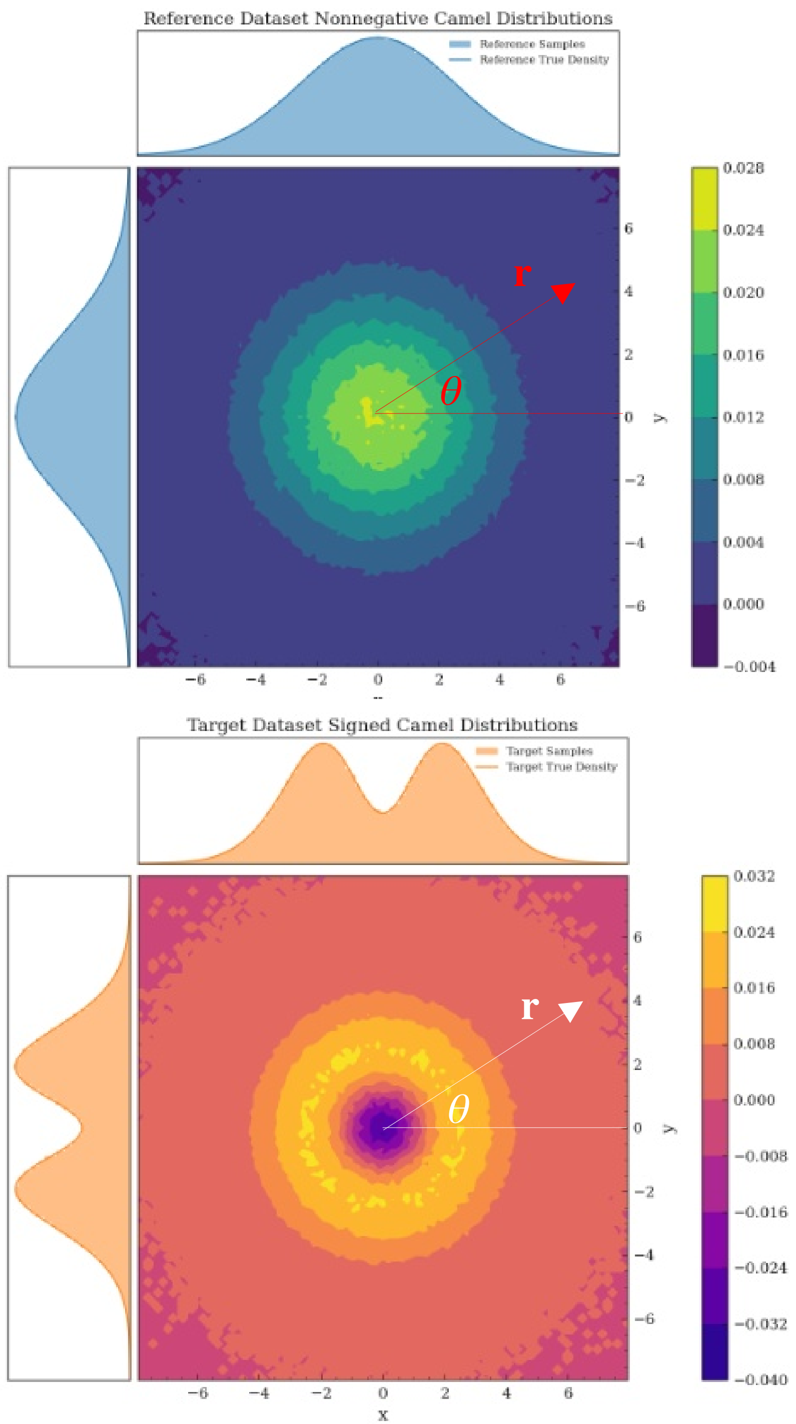
Per bin pull between the reference and the target

Distribution of pulls across all bins. Ideally want it to be Gaussian

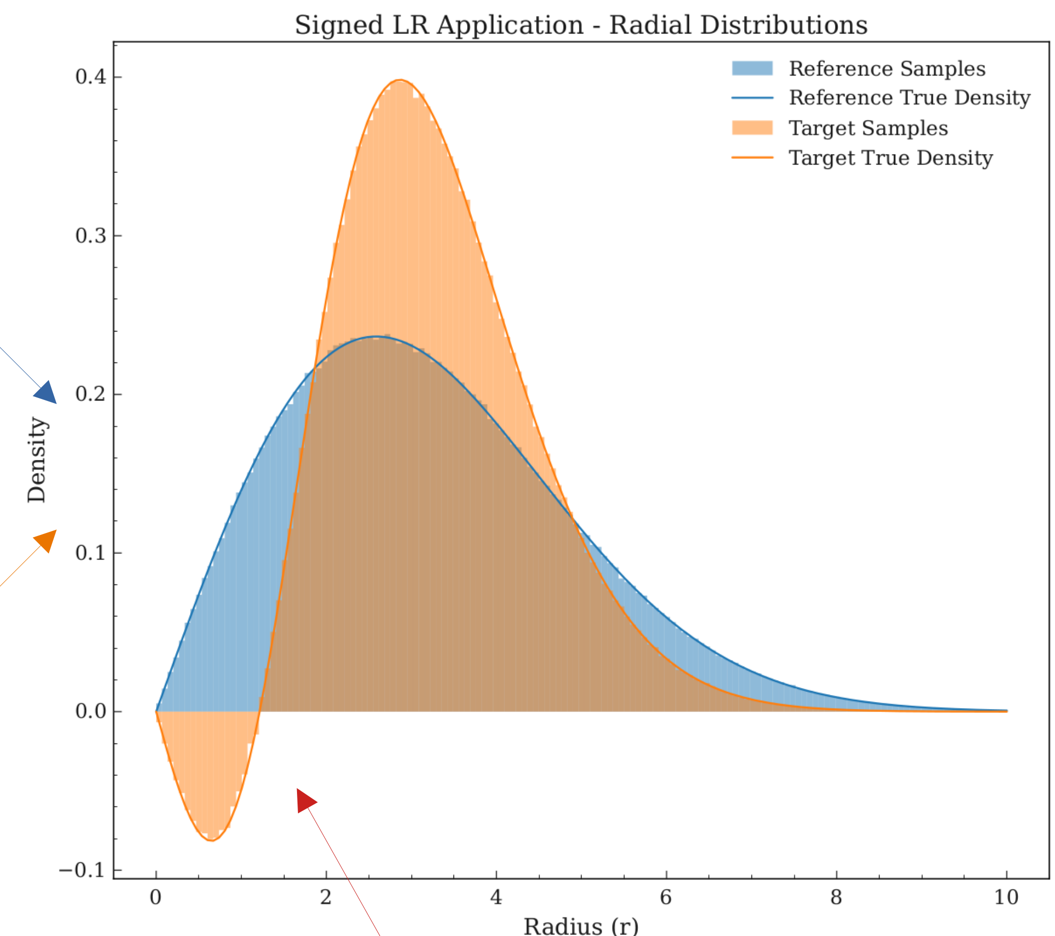
Toy Model: Observation space



Toy Model: Sample Space

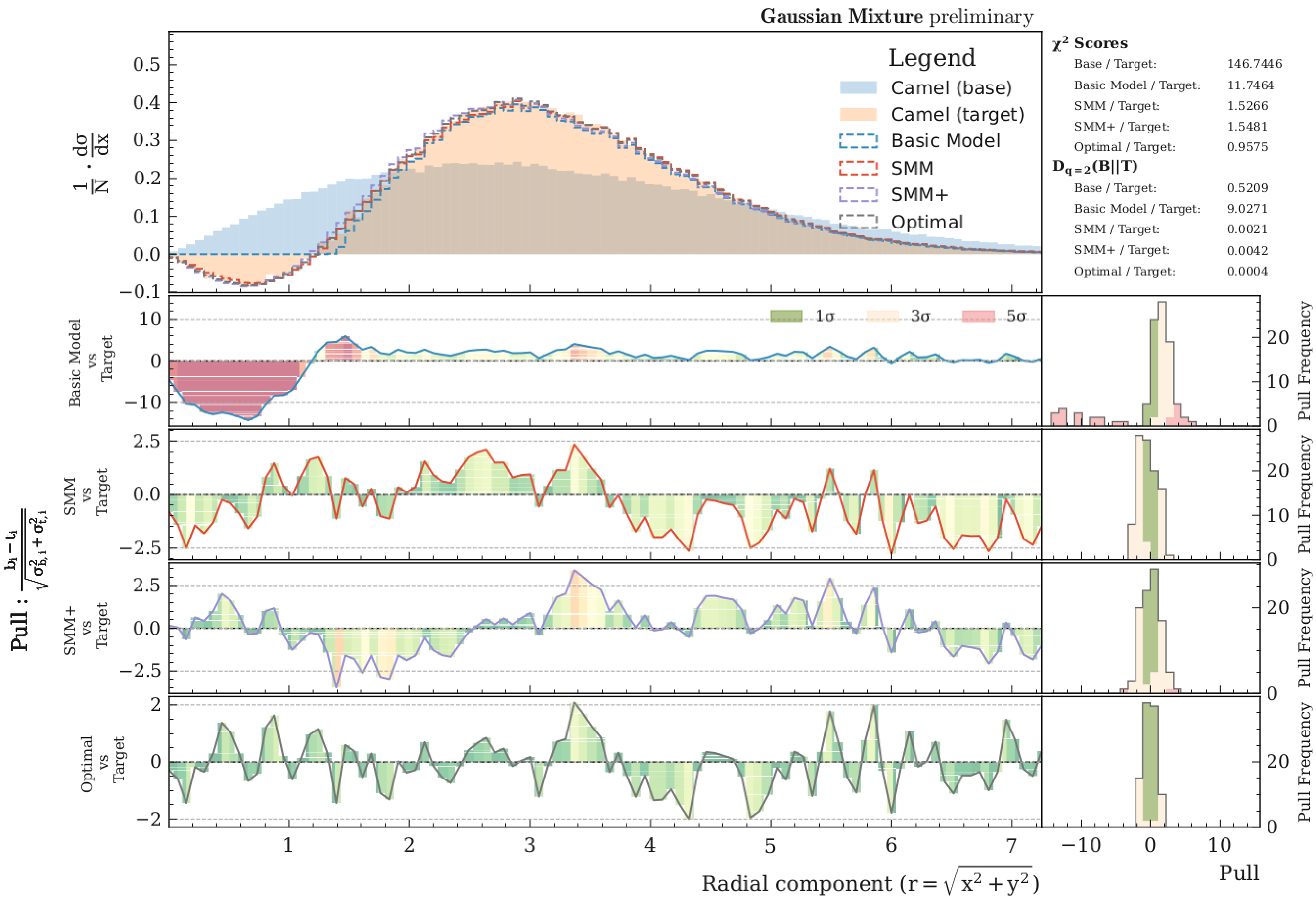


→ Characteristic shape of the distribution is entirely encoded in the radial polar co-ordinate '*r*' with angle '*θ*' uniformly distributed



→ In x/y co-ordinates the negative/positive parts are maginalised away concealing the purely -ve region

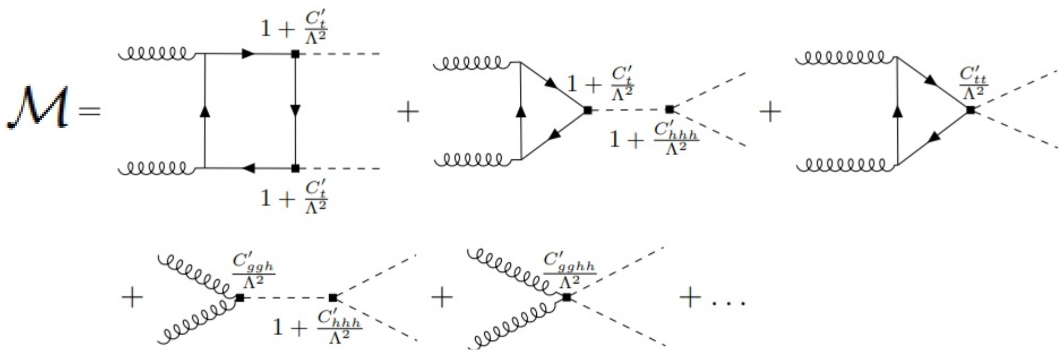
Toy Model: Sample Space



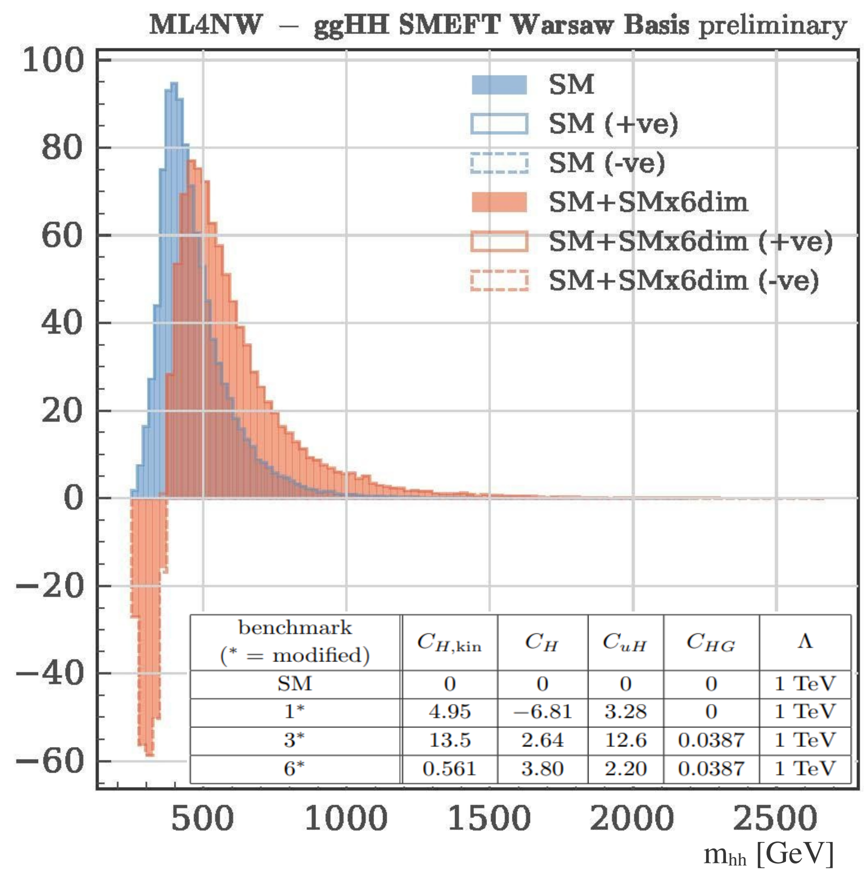
HEP: $gg \rightarrow HH$

SMEFT example: ggHH

→ **Synthetic data** generated for gg → *hh* process as an example domain adaptation problem:



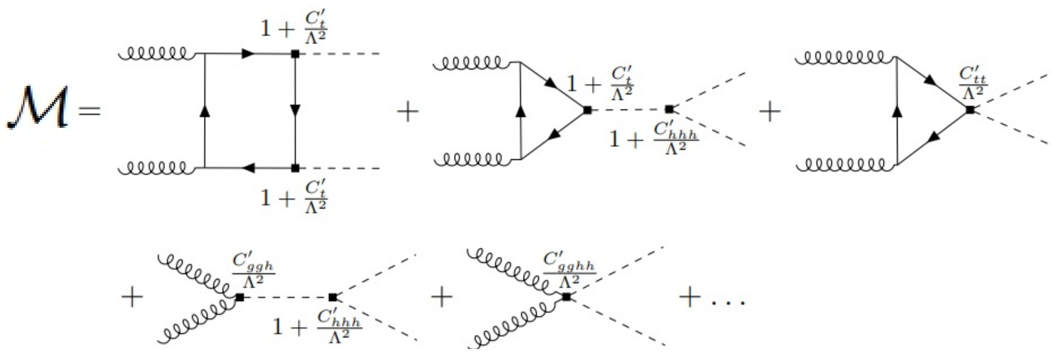
$$= \mathcal{M}_{\mathrm{SM}} + \mathcal{M}_{\mathrm{dim6}} + \mathcal{M}_{\mathrm{dim6}^2}\ ,$$



benchmark (* = modified)	$C_{H,\mathrm{kin}}$	C_H	C_{uH}	C_{HG}	Λ
SM	0	0	0	0	1 TeV
1*	4.95	−6.81	3.28	0	1 TeV
3*	13.5	2.64	12.6	0.0387	1 TeV
6*	0.561	3.80	2.20	0.0387	1 TeV

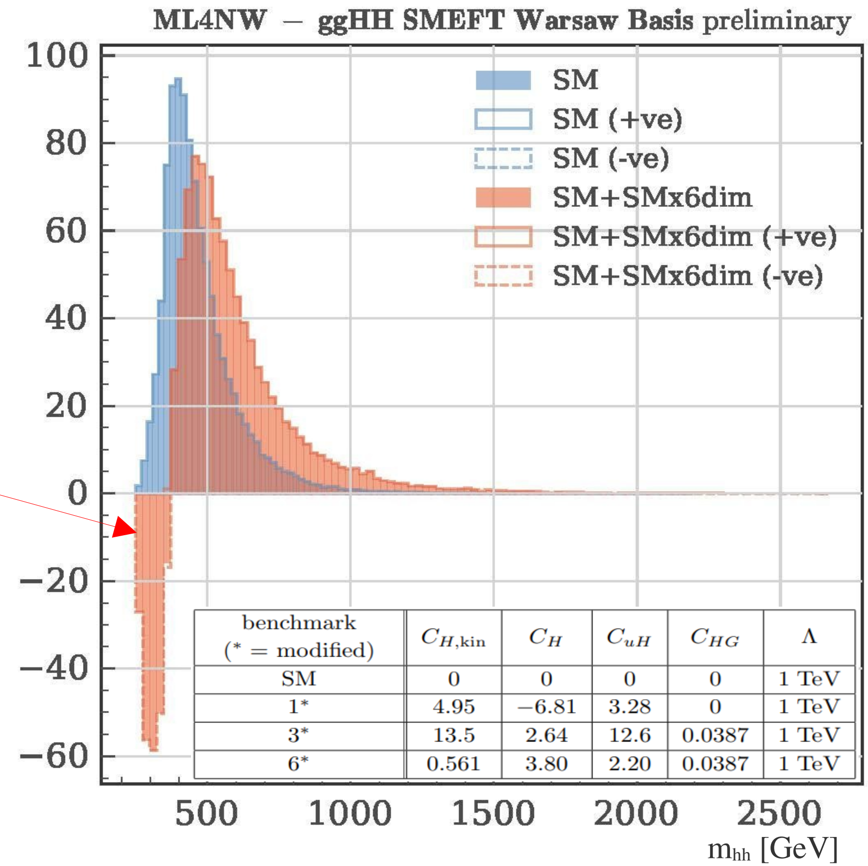
SMEFT example: ggHH

→ **Synthetic data** generated for gg → *hh* process as an example domain adaptation problem:



$$= \mathcal{M}_{\text{SM}} + \mathcal{M}_{\text{dim6}} + \mathcal{M}_{\text{dim6}^2} \,,$$

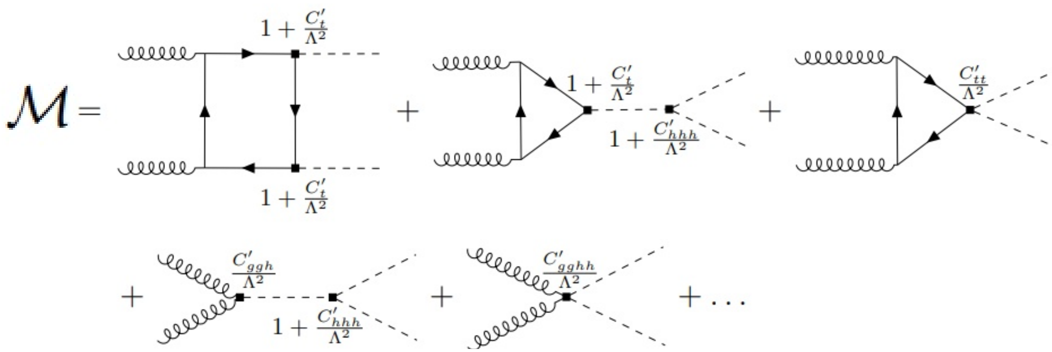
Quantum Intereference



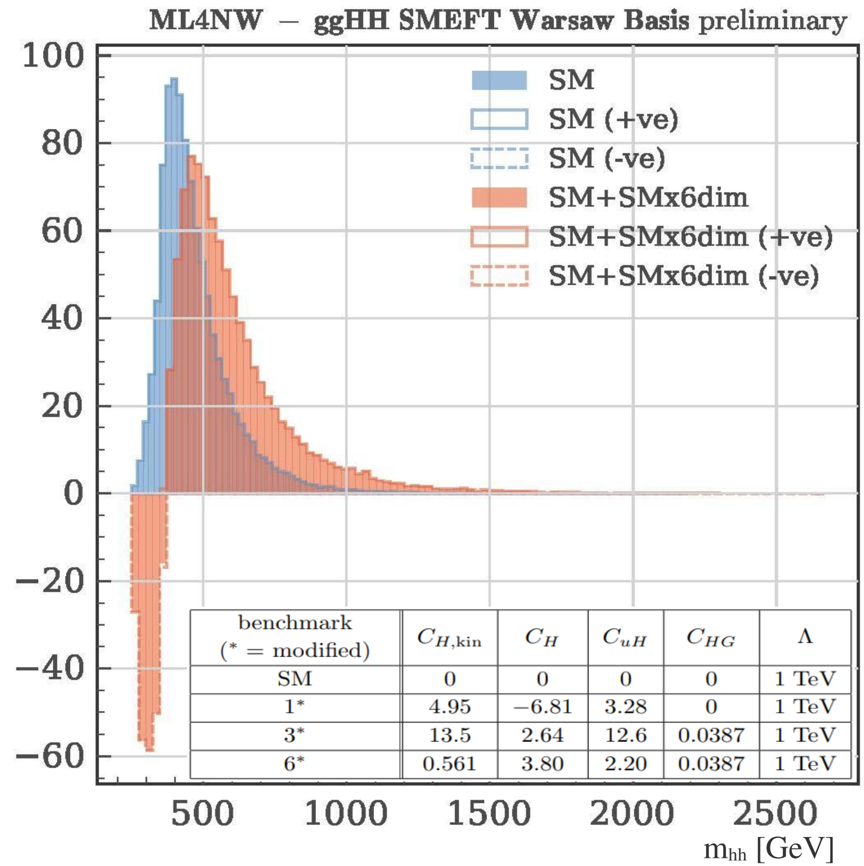
benchmark (* = modified)	$C_{H,\text{kin}}$	C_H	C_{uH}	C_{HG}	Λ
SM	0	0	0	0	1 TeV
1*	4.95	−6.81	3.28	0	1 TeV
3*	13.5	2.64	12.6	0.0387	1 TeV
6*	0.561	3.80	2.20	0.0387	1 TeV

SMEFT example: ggHH

→ **Synthetic data** generated for gg → *hh* process as an example domain adaptation problem:



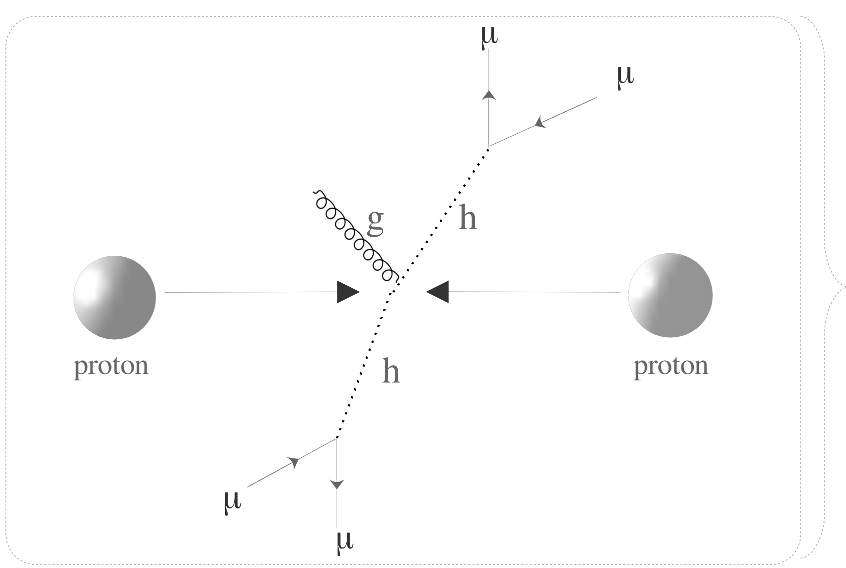
$$= \mathcal{M}_{\mathrm{SM}} + \mathcal{M}_{\mathrm{dim6}} + \mathcal{M}_{\mathrm{dim6}^2} \,,$$



benchmark (* = modified)	$C_{H,\mathrm{kin}}$	C_H	C_{uH}	C_{HG}	Λ
SM	0	0	0	0	1 TeV
1*	4.95	−6.81	3.28	0	1 TeV
3*	13.5	2.64	12.6	0.0387	1 TeV
6*	0.561	3.80	2.20	0.0387	1 TeV

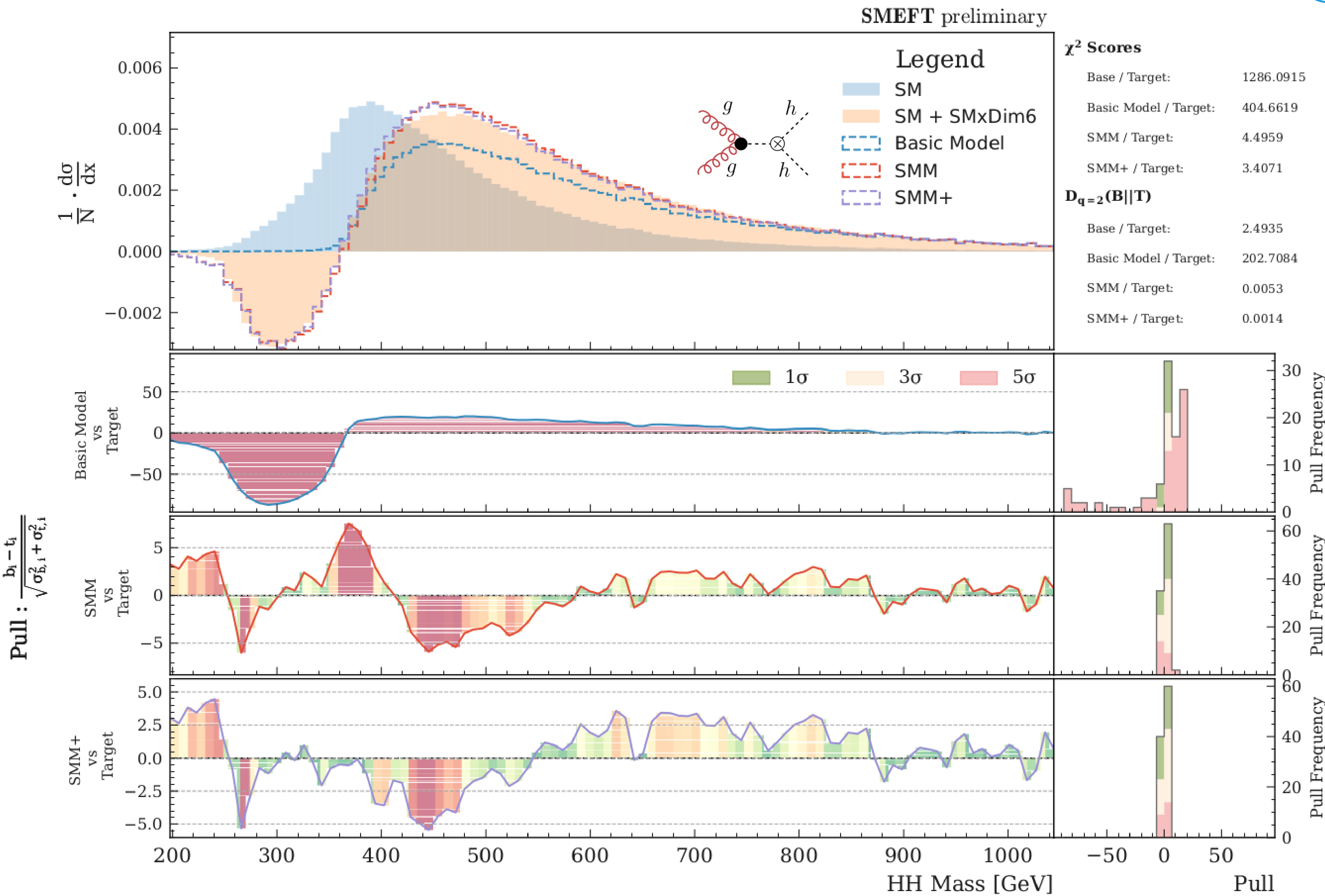
→ **Input domain,** given by the 4-vectors of the final state products:

gg → hh → 4μ + 1j



$\vec{x} \in \mathbb{R}^{16}$

ggHH: m_{hh} sample space



Conclusion

→ **Quasi-Probabilistic** neural likelihood ratio estimation (QNLRE):

- ✅ Decompose likelihood ratio problem using signed probability spaces to be quasi-probabilistic in nature:

$$r(x;\mathbf{c}) = \left[\frac{c_0 p_+(x|Y=0)}{c_1 p_+(x|Y=1)} + \frac{(1-c_0)p_-(x|Y=0)}{c_1 p_+(x|Y=1)}\right]^{-1} + \left[\frac{c_0 p_+(x|Y=0)}{(1-c_1)p_-(x|Y=1)} + \frac{(1-c_0)p_-(x|Y=0)}{(1-c_1)p_-(x|Y=1)}\right]^{-1}$$

- ✅ Mitigate -ve weight induced training variance in NLRE problems by casting -ve weighted data to positive domain $w_i \rightarrow |w_i|$:

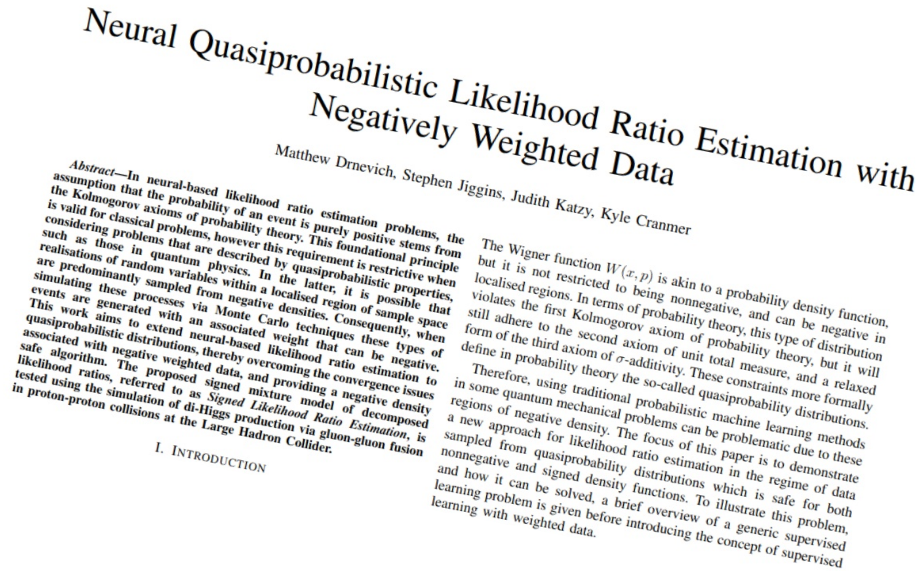
$$\mathrm{Var}_{\tilde{X},Y,W}\left(\theta_i^{t+1}\right) - \mathrm{Var}_{X,Y}\left(\theta_i^{t+1}\right) = \frac{\gamma^2}{N_{batch}}\mathbb{E}_{\tilde{X},Y,W}\left[(W^2 - W)\cdot\left(\frac{\partial}{\partial\theta_i}\bigg|_{\theta_i=\theta_i^t}\mathcal{L}(s(\tilde{X};\boldsymbol{\theta}^t),Y)\right)^2\right]$$

- ✅ New loss function to optimise QNLRE models that avoid divergences in the optimisation problem:

$$\mathbf{c}^* = \mathrm{argmin}_{\mathbf{c}\in\mathbb{R}^2}\mathbb{E}_{X,Y,W}\left[\left|1-\left(\frac{Y_0+Y_1\hat{r}(x;\mathbf{c})}{Y_0^2+Y_1^2\hat{r}(x;\mathbf{c})}\right)Y\right|^2 W\right]$$

→ **Examples of quasi-probabilistic** systems in HEP & beyond experiments:

- Heavy Neutral Higgs at the LHC - ATLAS-CONF-2024-001
- NLO SMEFT ggHH - 2204.13045
- Negative Probabilities in Financial Modeling - https://dx.doi.org/10.2139/ssrn.1773077



Neural Quasiprobabilistic Likelihood Ratio Estimation with Negatively Weighted Data

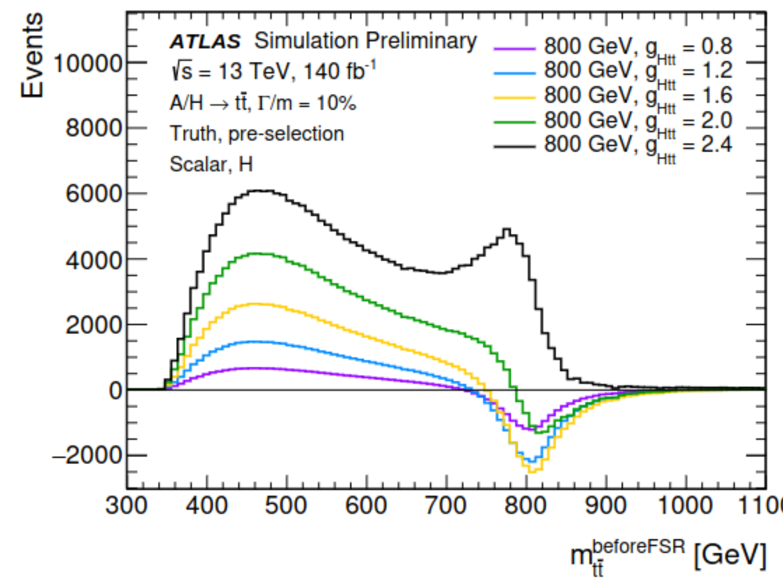
Matthew Drnevich, Stephen Jiggins, Judith Katzy, Kyle Cranmer

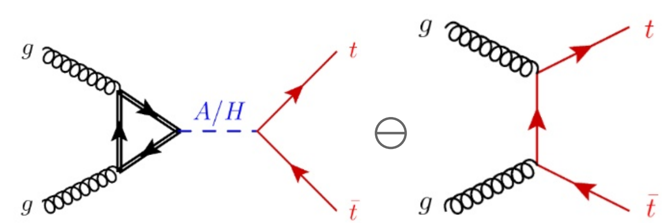
Abstract—In neural-based likelihood ratio estimation problems, the assumption that the probability of an event is purely positive stems from the Kolmogorov axioms of probability theory. This foundational principle is valid for classical problems, however this requirement is restrictive when considering problems that are described by quasiprobabilistic properties, such as those in quantum physics. In the latter, it is possible that realisations of random variables within a localised region of sample space are predominantly sampled from negative densities. Consequently, when simulating these processes via Monte Carlo techniques these types of events are generated with an associated weight that can be negative. This work aims to extend neural-based likelihood ratio estimation to quasiprobabilistic distributions, thereby overcoming the convergence issues associated with negative weighted data, and providing a negative density safe algorithm. The proposed signed mixture model of decomposed likelihood ratios, referred to as *Signed Likelihood Ratio Estimation*, is tested using the simulation of di-Higgs production via gluon-gluon fusion in proton-proton collisions at the Large Hadron Collider.

I. INTRODUCTION

The Wigner function $W(x,p)$ is akin to a probability density function, but it is not restricted to being nonnegative, and can be negative in localised regions. In terms of probability theory, this type of distribution violates the first Kolmogorov axiom of probability theory, but it will still adhere to the second axiom of unit total measure, and a relaxed form of the third axiom of σ-additivity. These constraints more formally define in probability theory the so-called quasiprobability distributions.

Therefore, using traditional probabilistic machine learning methods in some quantum mechanical problems can be problematic due to these regions of negative density. The focus of this paper is to demonstrate a new approach for likelihood ratio estimation in the regime of data sampled from quasiprobability distributions which is safe for both nonnegative and signed density functions. To illustrate this problem, and how it can be solved, a brief overview of a generic supervised learning problem is given before introducing the concept of supervised learning with weighted data.





Backup

Context: Likelihood Function

→ Likelihood function is key to the *scientific method:*

$$\mathcal{L}(\theta|x) = p_\theta(x) = P(X = x|\theta)$$

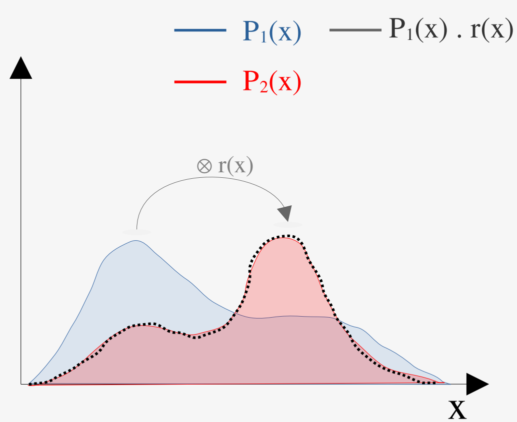
→ This is due to its prolific use in *statistical inferencing* problems via its ratio form:

$$r(x|\theta_1, \theta_2) = \frac{\mathcal{L}(\theta_1|x)}{\mathcal{L}(\theta_0|x)}$$

Neyman-Pearson Lemma

$$t_{\text{LR}} = -2 \cdot ln \left(\frac{\mathcal{L}(\theta_1|x)}{\mathcal{L}(\theta_0|x)} \right)$$

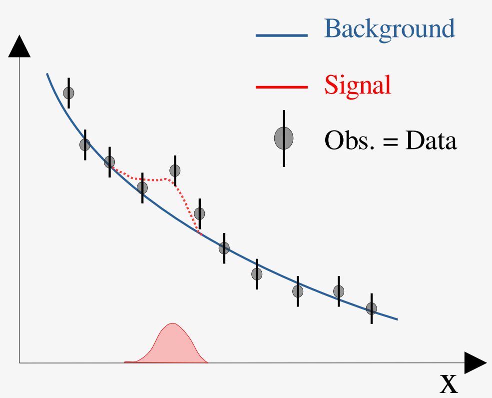
Domain Adaptation



Use the density ratio r(x) as a mapping function between domain spaces

r(x): $X_1 \rightarrow X_2$

Hypothesis Testing



Use the logarithm of the density ratio r(x) to calculate confidence in excluding hypothesis H_0 over H_1

Context: Likelihood Function

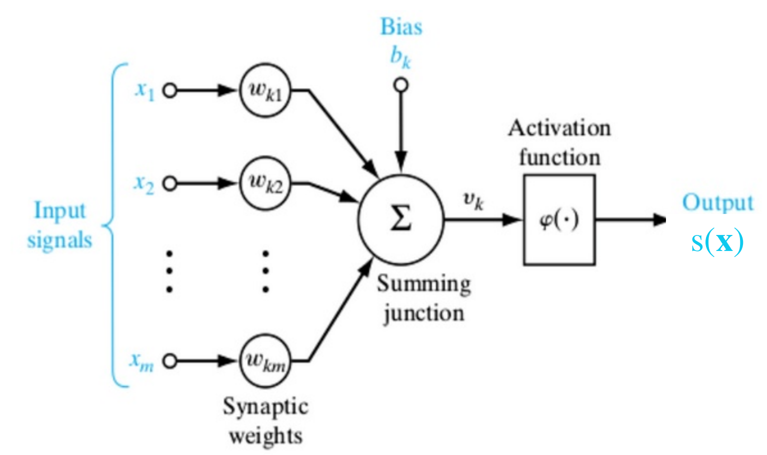
→ Likelihood function is key to the *scientific method:*

$$\mathcal{L}(\theta|x) = p_\theta(x) = P(X = x|\theta)$$

→ This is due to its prolific use in *statistical inferencing* problems via its ratio form:

$$r(x|\theta_1, \theta_2) = \frac{\mathcal{L}(\theta_1|x)}{\mathcal{L}(\theta_0|x)}$$

Neural Likelihood Ratio Estimation



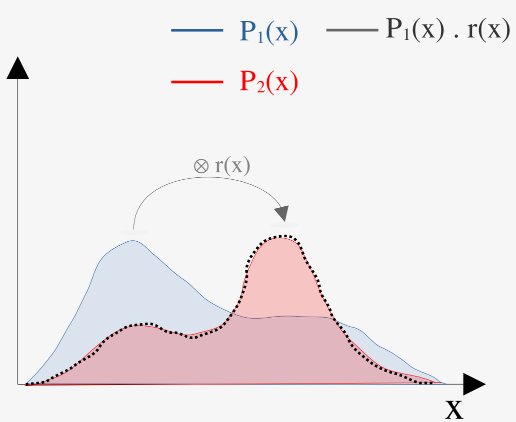
For a neural network with configurable set $\Phi = \{w_i\}^N_{i=1}$ parameters and a given loss functional $\mathcal{L}(s)$ of the form:

$$\mathcal{L}(s) = -\int_{\mathbb{R}^n} d^n x \left(p(x|\theta_0) \cdot A(s(x)) + p(x|\theta_1) \cdot B(s(x))\right)$$

The extrema ($\delta\mathcal{L}(s)/\delta s = 0$) of this general loss yields:

$$-\frac{A'(x)}{B'(x)} = \frac{p(x|\theta_1)}{p(x|\theta_0)} = r(x|\theta_0, \theta_1)$$

Domain Adaptation



Use the density ratio r(x) as a mapping function

r(x): $X_1 \rightarrow X_2$

Signed Probability Measures

→ **Decompose probability measure** into a signed measure:

$$\mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n \quad \Rightarrow \quad \mathbf{P}[X \in A] = \int_A q_X(\mathbf{x}) d\mathbf{x}^n = \int_A (p_+(\mathbf{x}) - p_-(\mathbf{x})) d\mathbf{x}^n$$

For a probability triplet $(\Omega, \mathcal{F}, P)$, one needs to show that the σ-additive function $\mu : \mathcal{F} \to \mathbb{R}$ can be a signed measure for any $E \in \mathcal{F}$:

$$\mu(E) = \mu^+(E) - \mu^-(E) \qquad (2)$$