DLScanner and LeStrat-Net: Machine learning for improved Monte Carlo exploration

Raymundo Ramos
Korea Institute for Advanced Study

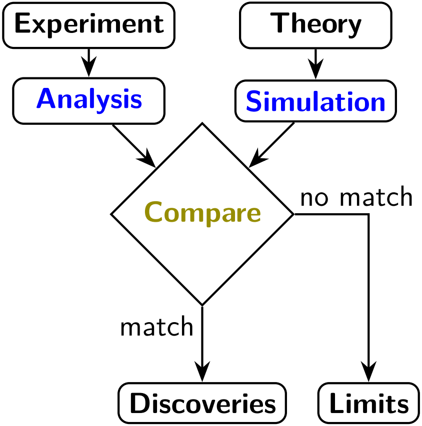
Based on:
JHEP 02 (2026) 077 by A. Hammad, RR, A. Chakraborty, P. Ko and S. Moretti
CPC 319 (2026) 109907 by M. Park and K. Ban and RR
CPC 314 (2025) 109659 by A. Hammad, RR

Particle Physics, Quantum Information And Machine Learning
IBS, Daejeon, South Korea
July 14, 2026

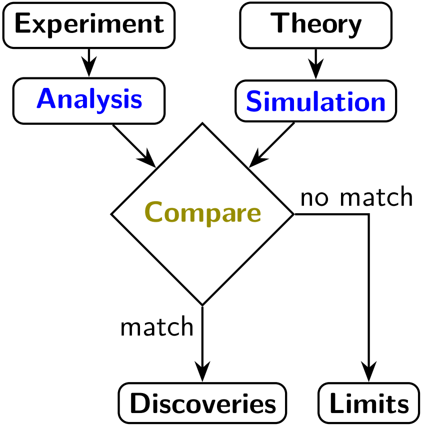
From theory to discovery (or limits)

Experiments grow in

- **complexity**,
- **precision**,
- **number**,
- ...



From theory to discovery (or limits)

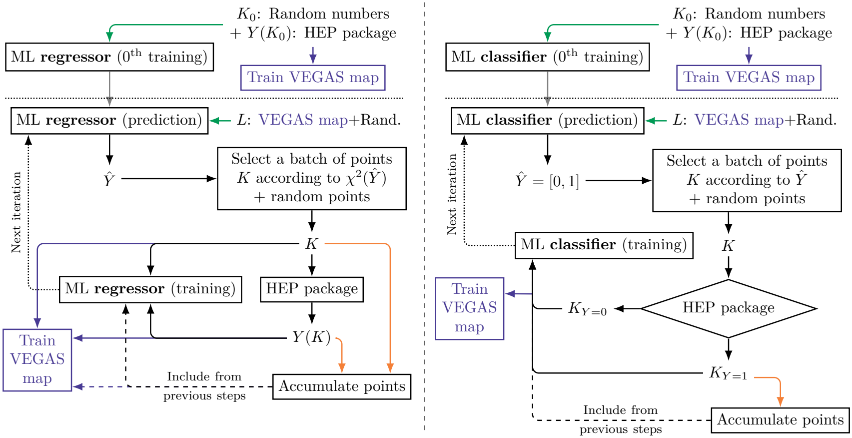


Simulations have to keep up

- **increasing dimensions**,
- **accuracy**,
- **many models**,
- **heavy calculations,**
- ...

Many opportunities for new (ML) techniques

Our detailed process [arXiv:2207.09959, arXiv:2412.19675]



`DLScanner`: a tool to streamline the process [arXiv:2412.19675]

pip install DLScanner

- Create Python function that runs your calculation.
- Formulate your output appropriately for either binary classification or regression.
- Set the size of the network and training parameters or define your own network.
- And many more options are available

DLScanner takes care of the rest.

Simple minimal example

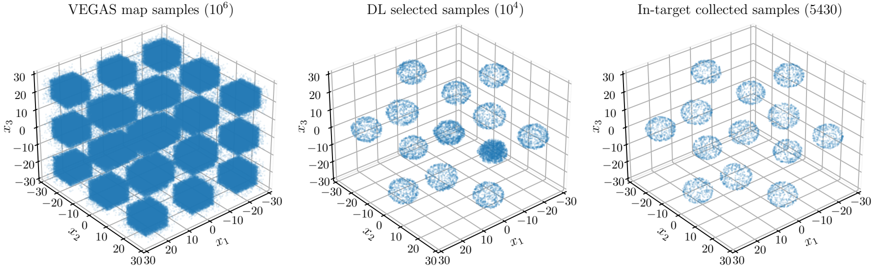
```
from DLScanner.gen_scanner import sampler
from user_module import user_function

ndim = 3
limits = [[-10, 10]]*ndim
method = "Classifier"
vegas_frac = 0.5

my_sampler = sampler(
    user_function, ndim, limits, method, vegas_frac=vegas_frac
)
```

```
steps = 10
my_sampler.advance(steps)
```

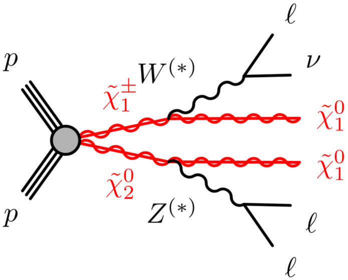
VEGAS maps

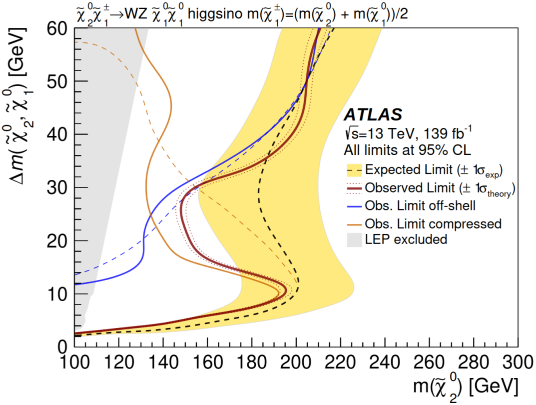


- Generate random points as close as possible to the target
- Increase number of generated points when target regions are small.
- Shortcomings of VEGAS maps are corrected by the network prediction.

DLScanner and hints new physics [JHEP 02 (2026) 077]

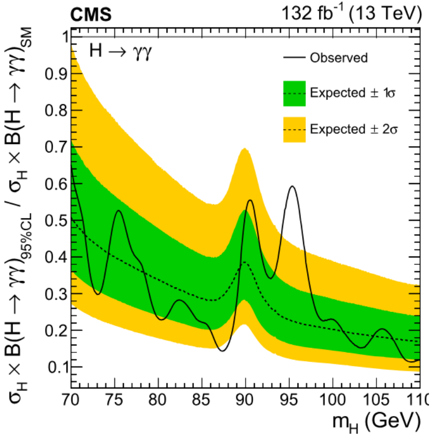
Electroweakino searches [ATLAS 2106.01676]



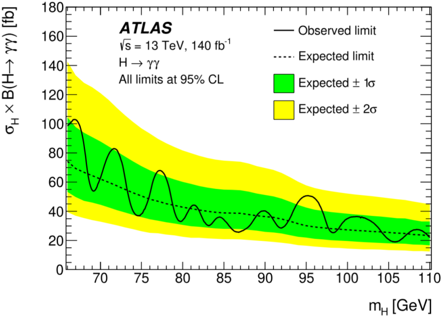


DLScanner and hints new physics [JHEP 02 (2026) 077]

Hints of a $\sim$**95.4 GeV scalar**



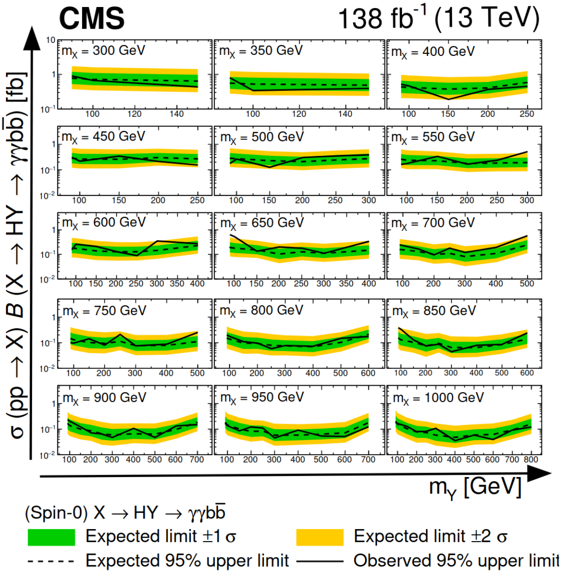
2.9σ local (1.3σ global)
[2405.18149]



1.7σ local
[2407.07546]

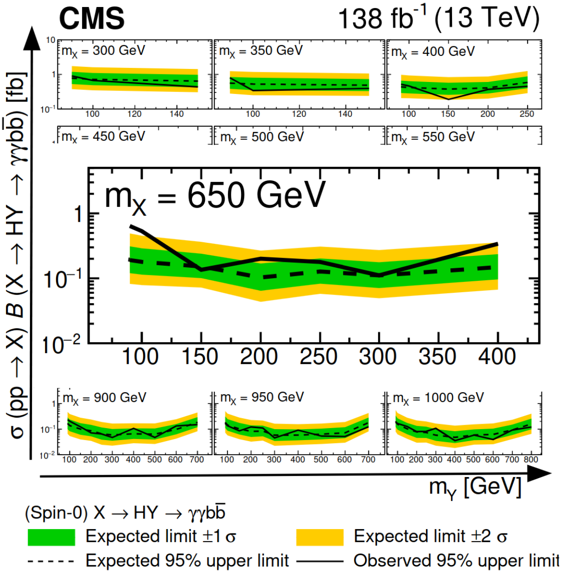
DLScanner and hints new physics [JHEP 02 (2026) 077]

Hints of scalars $\sim$**650 GeV** $\to$ 90 GeV (loc. 3.8σ) [CMS 2310.01643]



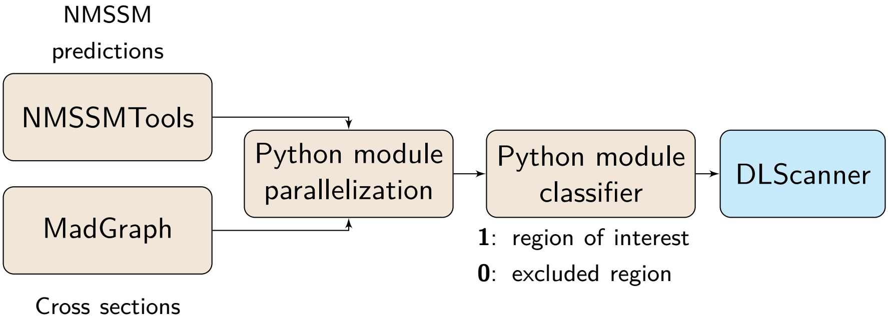
DLScanner and hints new physics [JHEP 02 (2026) 077]

Hints of scalars $\sim$**650 GeV** $\rightarrow$ 90 GeV (loc. 3.8σ) [CMS 2310.01643]



Setup of the DL scan

Calculations use `NMSSMTools` which coordinates a series of tools: `NMHDECAY`, `NMSPEC`, `NMGMSB`, `NMHDECAY_CPV`, `NMSDECAY`, `micrOMEGAs` and `SModelS`.



Setup of the DL scan

$$W_{\text{NMSSM}} = W_{\text{MSSM}} + \lambda \hat{H}_u \hat{H}_d \hat{S} + \frac{1}{3}\kappa \hat{S}^3,$$

$$V_{\text{soft}} = m^2_{H_u}|H_u|^2 + m^2_{H_d}|H_d|^2 + m^2_S|S|^2 + \lambda A_\lambda \hat{H}_u \hat{H}_d \hat{S} + \frac{1}{3}\kappa A_\kappa \hat{S}^3,$$

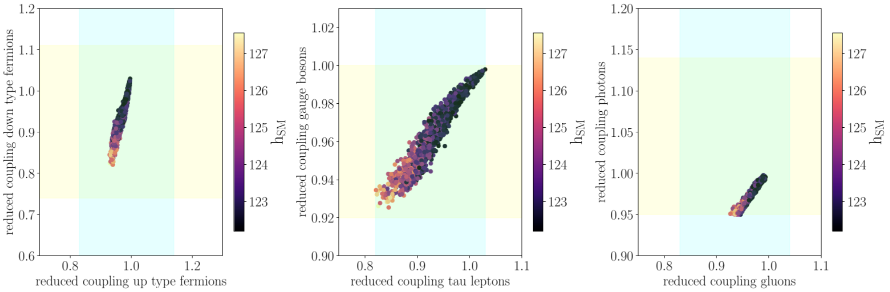
Preliminary scan in **wide** range.
Definitive scan in **narrow** range.

	$\tan\beta$	λ	κ	A_λ
wide	$[1.97,\ 10.9]$	$[0.013,\ 0.687]$	$[0.0058,\ 0.391]$	$[-5000,\ 480]$
narrow	$[3.2,\ 6.2]$	$[0.07,\ 0.42]$	$[0.05,\ 0.3]$	$[351,\ 834]$
	A_κ	μ_{eff}	M_1	M_2
wide	$[-621,\ 362]$	$[-244,\ 291]$	$[178,\ 3000]$	$[304,\ 10000]$
narrow	$[-300,\ -150]$	$[120,\ 220]$	$[500,\ 3000]$	$[750,\ 10000]$
	M_3	A_t	M_{Q_3}	M_{U_3}
	$[423,\ 5000]$	$[-5000,\ 1288]$	$[272,\ 10000]$	$[570,\ 10000]$

The constraints: SM-like Higgs properties

$m_{h125} = 125.2 \pm 3$ GeV due to theoretical uncertainty

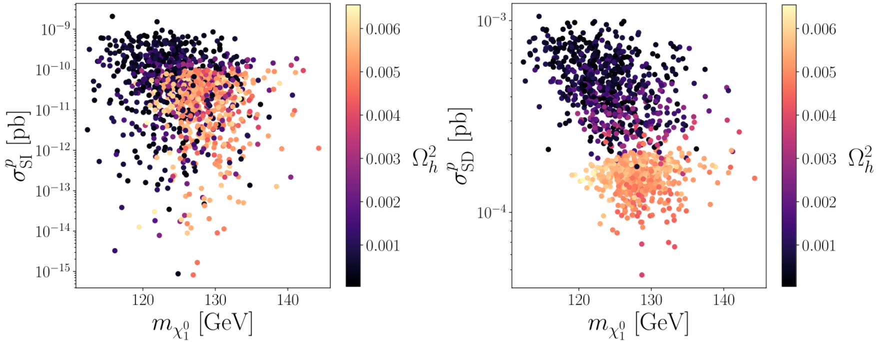
Higgs reduced couplings



The constraints: Dark matter measurements

$$\Omega_{\mathrm{CDM}} h^2 < 0.122$$

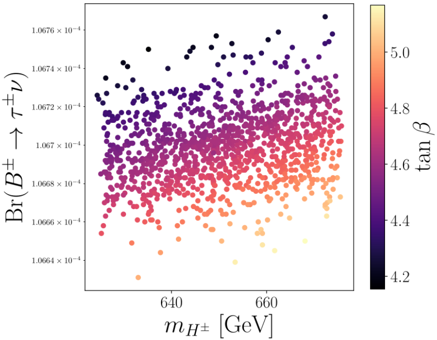
σ^p_{SI} below LZ limits [2207.03764]

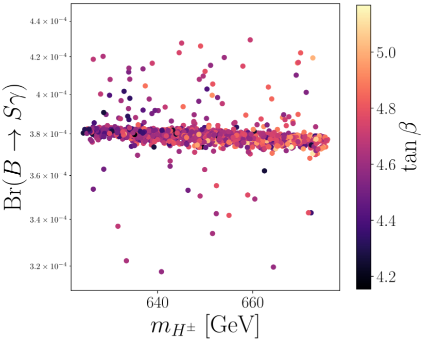


Relic density consistent with other studies: Ellwanger and Hugonie [2309.07838]

The constraints: Low energy observables

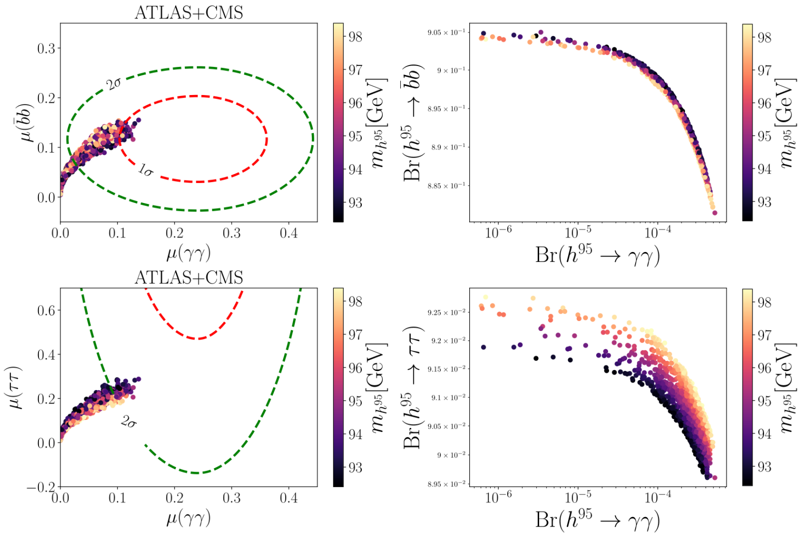
$$\mathrm{BR}(B \to X_s\gamma)_{E_\gamma \geq 1.6\ \mathrm{GeV}} = (3.32 \pm 0.15) \times 10^{-4}\,,$$
$$\mathrm{BR}(B^\pm \to \tau^\pm \nu_\tau) = (1.11 \pm 0.165) \times 10^{-4}\,.$$



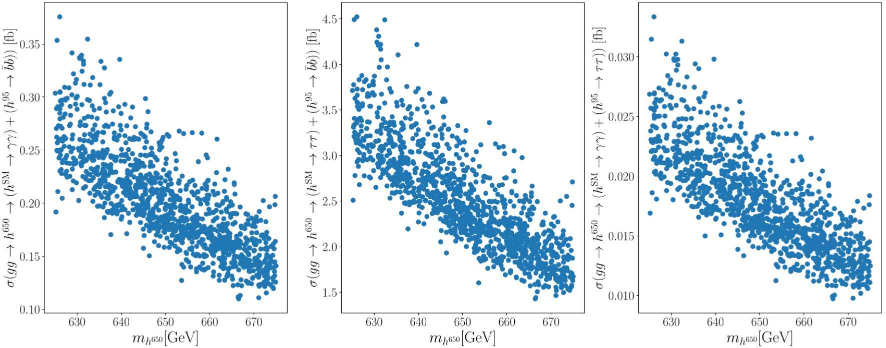


Finding a 95 GeV scalar

$$\mu_{\gamma\gamma} = \frac{\sigma(gg \to h_{95} \to \gamma\gamma)}{\sigma(gg \to h_{95}^{\mathrm{SM}} \to \gamma\gamma)}\,, \qquad \mu_{b\bar{b}} = \frac{\sigma(e^+e^- \to Zh_{95} \to Zb\bar{b})}{\sigma(e^+e^- \to Zh_{95}^{\mathrm{SM}} \to Zb\bar{b})}, \qquad \mu_{\tau^+\tau^-} = \frac{\sigma(gg \to h_{95} \to \tau^+\tau^-)}{\sigma(gg \to h_{95}^{\mathrm{SM}} \to \tau^+\tau^-)}\,.$$



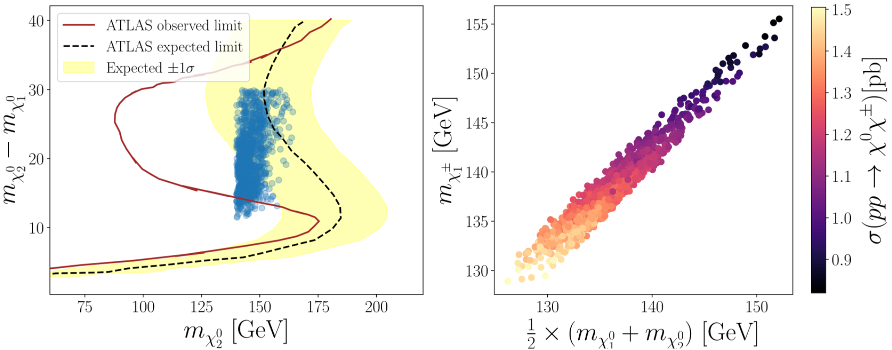
Finding a 650 GeV scalar



Remember that local significances are still low in the 2-3σ range.

Finding electroweakinos

Get EWino in the discrepant region



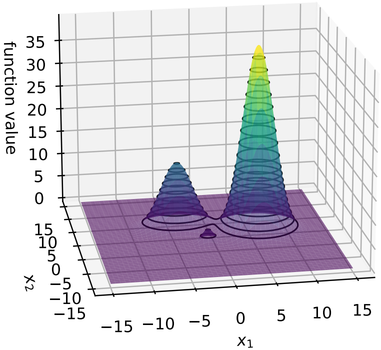
For more details check: Explaining data excesses over the NMSSM parameter space with Deep Learning techniques by A. Hammad, RR, A. Chakraborty, P. Ko and S. Moretti, JHEP 02 (2026) 077

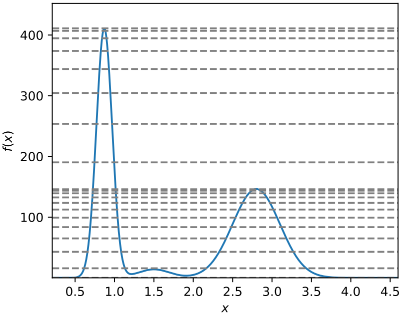
Part II

`LeStrat-Net`: Lebesgue style stratified sampling

LeStrat-Net: Lebesgue style stratified sampling

Define regions limited by values of the integrand



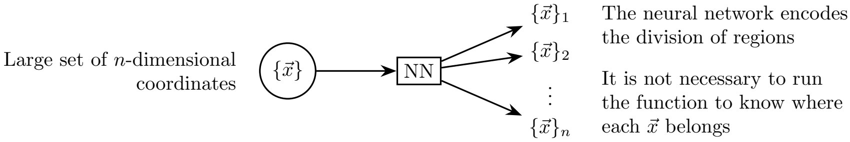


$$\Phi_j = \{x \mid l_j < f(x) \leq l_{j+1}\} \quad \rightarrow \quad E(I) = \sum_j E(V_{\Phi_j}) E(\langle f \rangle_{\Phi_j})$$

Remixing stratified sampling with a neural network

- Neural networks (NN) as generic function approximators
- Useful when training a NN **could be more efficient** than passing every single point through a heavy calculation

Main idea: train the NN to classify points according to contours



Evaluations of the neural network $\rightarrow$ determine $E(V_{\Phi_j})$, reduce $\sigma^2_{\Phi}(V_{\Phi_j})$

Remixing stratified sampling with a neural network

$$\sigma^2(E(I)) = E^2(V_{\Phi_j})\sigma^2_\Phi(\langle f\rangle_{\Phi_j}) + E^2(\langle f\rangle_{\Phi_j})\sigma^2_\Phi(V_{\Phi_j}) + \sigma^2_\Phi(\langle f\rangle_{\Phi_j})\sigma^2_\Phi(V_{\Phi_j})$$

- $E^2(V_{\Phi_j})\sigma^2_\Phi(\langle f\rangle_{\Phi_j})$

$\sigma^2_\Phi(\langle f\rangle_{\Phi_j})$: **reduced** by partitioning, **limited** by contours of $f(x)$.
Only part where the number of evaluations of $f(x)$ is important
Inaccurate or inconsistent partitioning increases variance.

- $E^2(\langle f\rangle_{\Phi_j})\sigma^2_\Phi(V_{\Phi_j})$

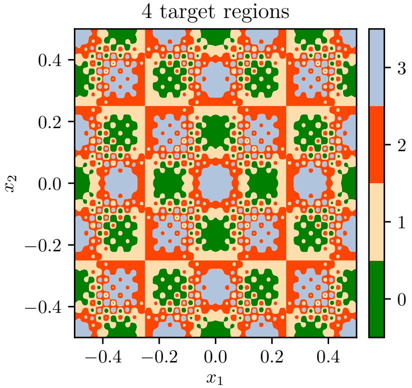
$\sigma^2_\Phi(V_{\Phi_j})$ reduced by evaluations of neural network.
Many techniques can be used to reduce this.
Evaluating the network is fast.

- $\sigma^2_\Phi(\langle f\rangle_{\Phi_j})\sigma^2_\Phi(V_{\Phi_j})$

This shrinks faster than the other two

Complicated regions and network size

$$f_{\mathsf{osc}}(x_1, x_2) = \prod_{j=1}^{2} \sin(40\pi x_j)\sin(4\pi x_j) + \prod_{j=1}^{2} \cos(6\pi x_j),$$

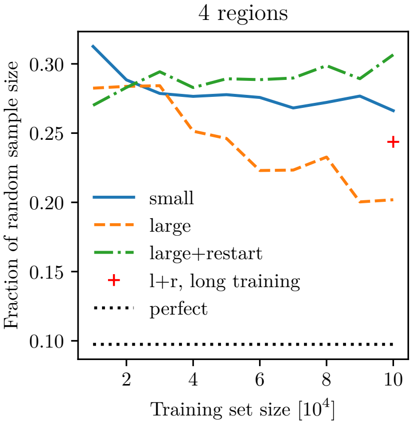


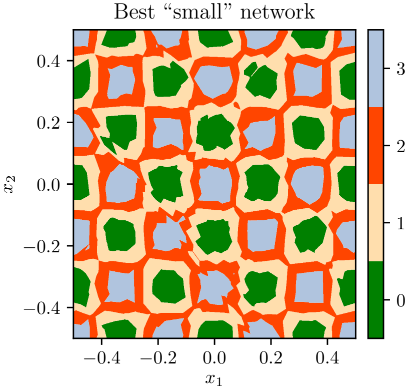
$$\begin{aligned}
\Phi_0 &: \quad f_{\mathsf{osc}} \leq -0.5\,,\\
\Phi_1 &: \quad -0.5 < f_{\mathsf{osc}} \leq 0\,,\\
\Phi_2 &: \quad 0 < f_{\mathsf{osc}} \leq 0.5\,,\\
\Phi_3 &: \quad f_{\mathsf{osc}} > 0.5\,.
\end{aligned}$$

- Many disconnected subregions
- Difficult contours

Complicated regions and network size

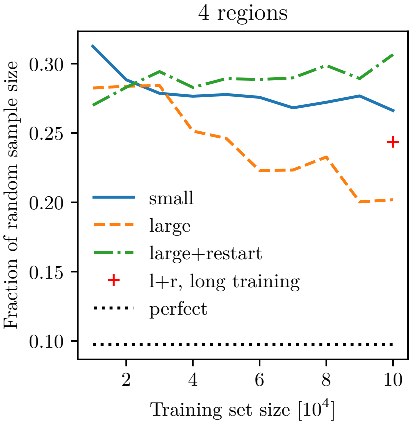
Fraction of random samples: $$\frac{N_{\text{strat}}}{N_{\text{single}}} = \frac{n_{\text{reg}}}{V_\Phi^2 \sigma_\Phi^2(f)} \sum_j V_{\Phi_j}^2 \sigma_{\Phi_j}^2(f)$$

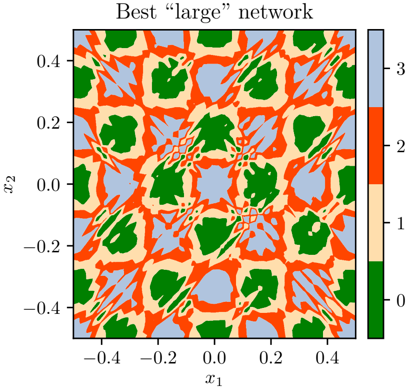




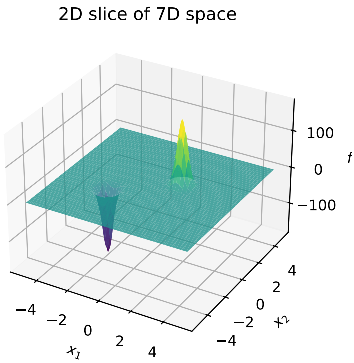
Complicated regions and network size

Fraction of random samples: $$\frac{N_{\mathsf{strat}}}{N_{\mathsf{single}}} = \frac{n_{\mathsf{reg}}}{V_\Phi^2 \sigma_\Phi^2(f)} \sum_j V_{\Phi_j}^2 \sigma_{\Phi_j}^2(f)$$





Integration: 7D function with large cancellation



normalized $f_+(x)$ $f_-(x)$ $f_{\mathrm{bg}}(x)$

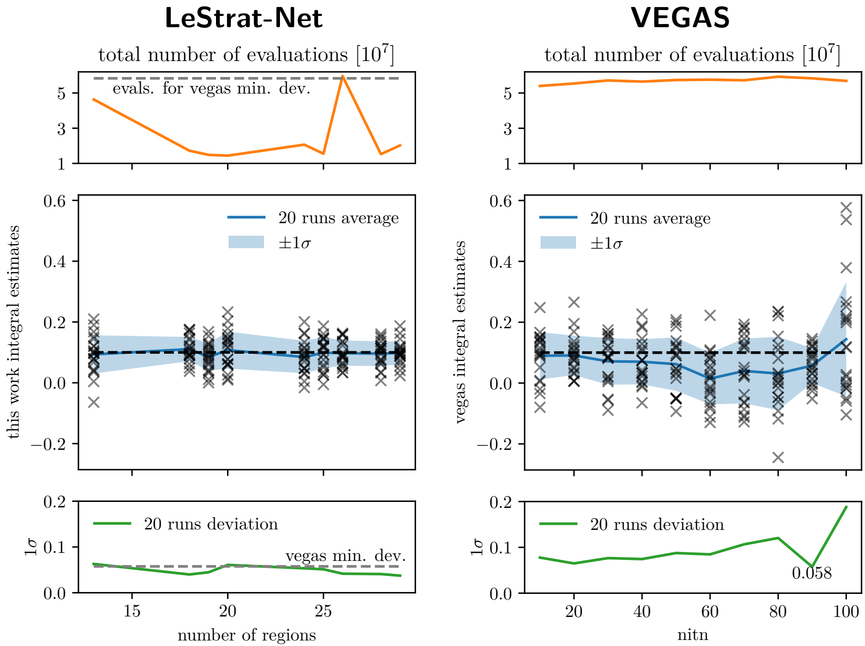
$$x \in [-5,5]^7$$

$$f(x) = 100[f_+(x) - f_-(x)] + 0.1 f_{\mathrm{bg}}(x)$$

$$\int f(x)dx = \int 0.1 f_{\mathrm{bg}}(x)dx \approx 0.1$$

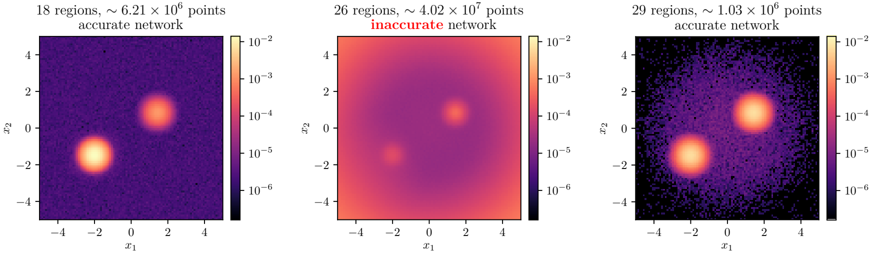
Multilayer perceptron: 2 hidden layers, 7D input

- Hidden: nodes $2 \times n_{\text{reg}} \times 7$, $n_{\text{reg}} \times 7$, activation ReLU
- Output: $n_{\text{reg}} - 1$, activation: *tanh*


LeStrat-Net
VEGAS
total number of evaluations $[10^7]$
evals. for vegas min. dev.
20 runs average
$\pm 1\sigma$
this work integral estimates
vegas integral estimates
20 runs deviation
vegas min. dev.
1σ
0.058
number of regions
nitn

Integration: 7D function with large cancellation

2D density of points for last integration step



18 regions	26 regions	29 regions
6.21×10^6 points	4.02×10^7 points	1.03×10^6 points
accurate net	**inaccurate** net	accurate net

(Accurate: misclassification by 2 regions is negligible)

Event generation: Quark pair to electron + positron

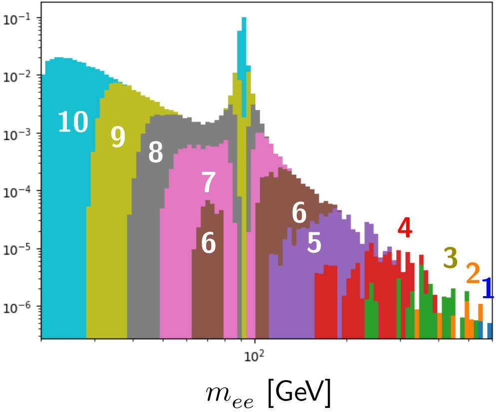
Very simple example:

$$u\bar{u} \rightarrow e^- e^+$$

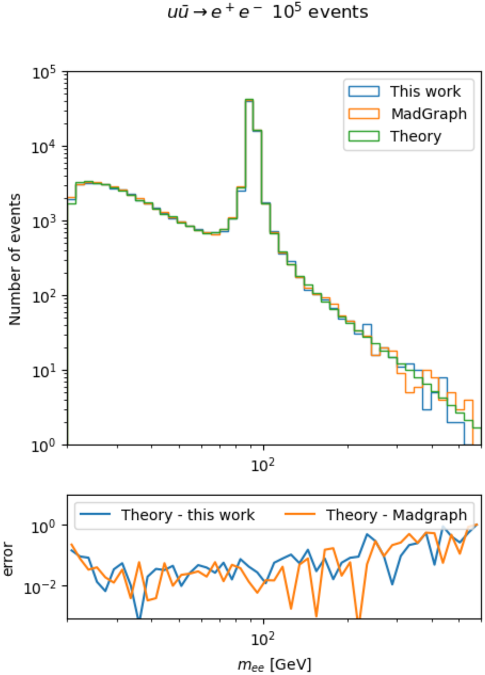
- `ROOT` - `TGenPhaseSpace`: phase space generator.
- `Madgraph` (standalone mode): matrix element.
- `NNPDF23`: parton density function.
- cuts: leptons: $p_T > 10\,\text{GeV}$, $|\eta| < 2.5$

Generate events: 10 usable regions

e^-e^+ invariant mass projection



- Sample each region until enough events are accumulated. **NN can tell which regions points belong to**.
- Select points using correct result.



- 10^5 unweighted events
- High m_{ee} error expected from thinning of sample.
- Invariant mass around Z resonance is similar when comparing to MadGraph
- Unweighting **efficiency increases with more regions**.

⚠ NOTE: **more regions** require **more points** for training

Summary

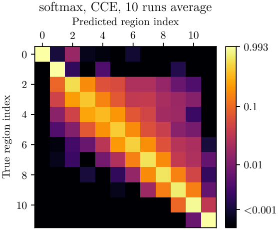
- Monte Carlo simulations could be challenging due to
  - $$ Time consuming costly operations
  - ❃ Complicated characteristics of the problem
- ➔ Use a neural network to aid in analysis of domain or parameter space
- ➔ Neural network can efficiently predivide regions that may be important
- ➔ Different processes can be formulated to let the neural network take care of resource consuming parts.
  - Concentrate on high importance regions
  - Reduce work in regions that contribute less to results

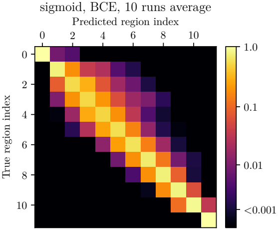
Thanks for listening!

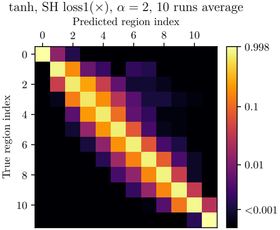
BACK UP

Remixing stratified sampling with a neural network

Next question: How to communicate network output and region labels?
(Test: 12 regions in simple conical function with 6D base)







multiclass	multilabel		
label by region index	**one label for each limit (above or below)**		
softmax categorical crossentropy	sigmoid binary crossentropy	or	tanh squared hinge

(Including tests with loss weighted with misclassification distance)