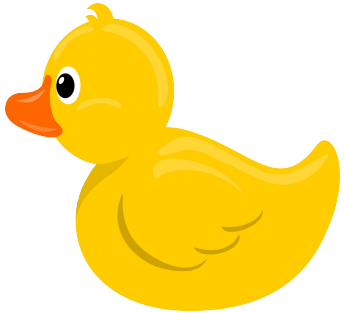


Quasi Anomalous Knowledge
UW EPE ML Meeting (Nov 29, 2021)

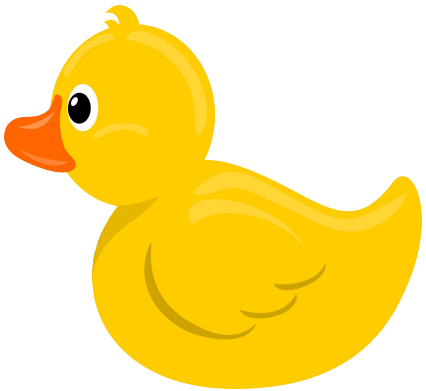
Sangeon Park, Dylan Rankin
Mikaeel Yunus, Silviu Udrescu, Philip Harris (MIT)

Overview

QUAK : QUasi Anomalous Knowledge

Today I will focus on Key Ideas of QUAK & the ML techniques that made it possible

1. **Brief intro Anomaly Detection in HEP**
2. ML techniques in QUAK + Technical Details
3. Key ideas used in QUAK
4. QUAK in Action (Signal Extraction Strategies)
5. Outlook



Why anomaly detection?

Strong motivation that there must be Beyond Standard Model (BSM) physics

Dark matter/energy, origin of neutrino mass, and so on

No BSM physics found at the LHC with searches targeting specific models

After the discovery of the Higgs (2012), effort focused on model specific searches

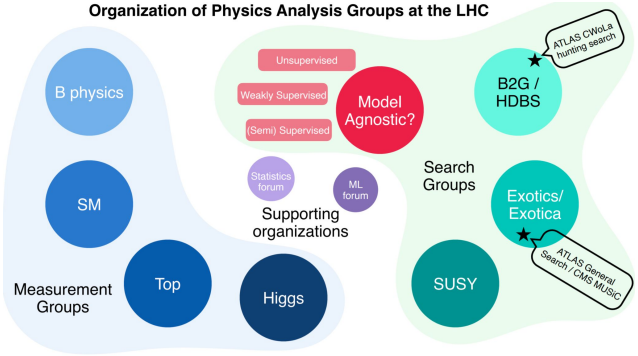
Supersymmetry, extra dimensions, extended Higgs and many more, but no convincing evidence yet

We need to start thinking about **model agnostic searches** : “anomaly detection”

Anomaly detection at colliders = searches not targeting specific models

Use ideas from “anomaly detection” in applied ML, but have to solve problems unique to HEP

2101.08320



Community Wide Effort

We need to have multiple strategies

Different methods will have different sensitivities to different **regions of search space** and **S/B ratio**

Big **community wide efforts** to come up with a wide variety of search strategies

LHC Olympics (2020) = challenge with 3 “black box” datasets with hidden embedded signal

- 18 algorithms submitted (2101.08320)

DarkMachines Challenge (2021) = challenge to detect a wide ensemble of signals

- 16 algorithms submitted (2105.14027)

34 algorithms + a lot more, all with unique approaches!

A wide variety of ideas, guiding principles, choice of input features and training set

Different ways to categorize approaches

What does the main algorithm do?

Dimensionality Reduction / Density Estimation / Overdensity / Clustering ...

PCA, AE, VAE, Flows, deep sets, noisy labels, isolation forest, k-means, BDT

What kind of input is used?

Images (CNN) / Particles (RNN, Graphs) / Processed Tabular Variables (MLPs)

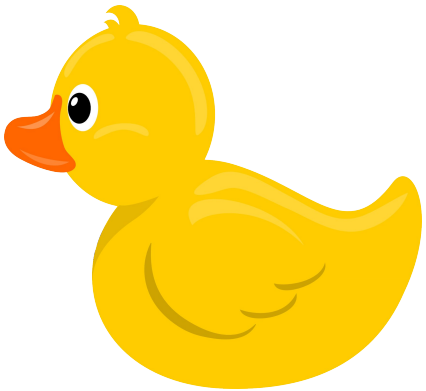
Is the algorithm trained on data or simulation?

How much signal information is used?

Unsupervised / Weakly Supervised / Semi-Supervised

Overview

1. Brief intro Anomaly Detection in HEP
2. **ML techniques in QUAK + Technical Details**
3. Key ideas used in QUAK
4. QUAK in Action (Signal Extraction Strategies)
5. Outlook



Density estimation based searches

We built QUAK with density estimation algorithm

Density estimation based methods are **most common** - 26/34 methods

Many ways to do it: kernel methods, VAEs, Flows, GANs etc..

Here you want the model to estimate the probability distribution of high dimensional data

Typical search works in a following way:

1. Train on background events(SM) a model to learn the distribution of the background
2. In testing select events with low p_{bkg}

Density estimation

Problem : learn $p(x)$ from data, where x is in some high dimensional space

For anomaly detection, Learn p_{bkg}, choose events with low p_{bkg}

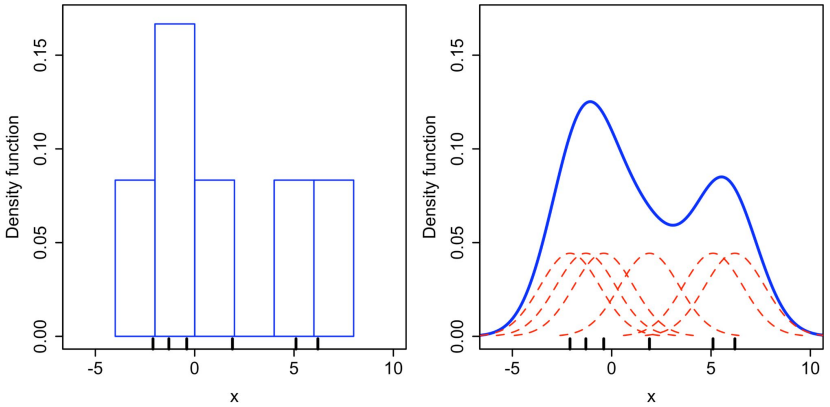
Classic non-parametric methods: histograms, kernel density estimation

$O\left(n^{-4/4+d}\right)$ Convergence rate

very very slow for high dimensional data: *Curse of Dimensionality*

(Limits around 3-6 dimension)

Used to be **very hard for high dimensional data**



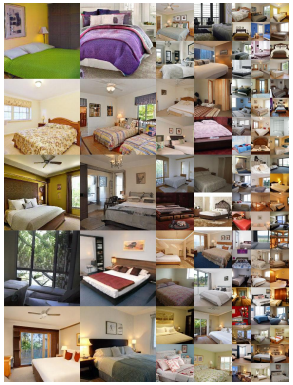
Deep learning to the rescue

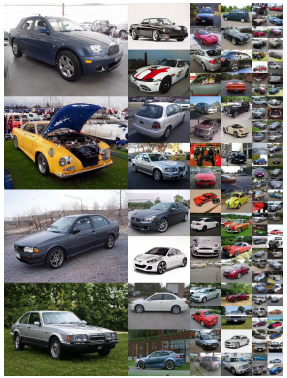
Explosion in deep generative models starting with GANs and VAEs

Huge advancement in latent variable based models last few years

If you learn the distribution of data, you can also sample from it to generate data

You can make realistic looking face / pictures / synthetic data/ etc.





Density estimation with neural networks

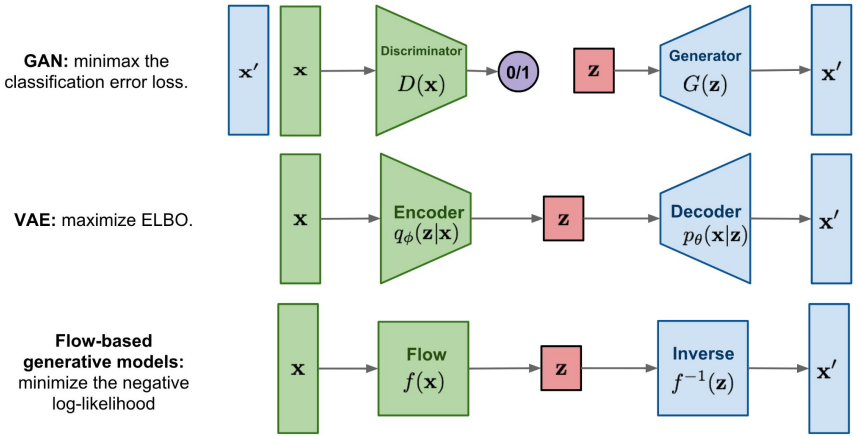
Parameter Inference / Generation are other sides of the same coin : You learn the density of the data distribution

Adversarial Training(implicit density)

GAN

Latent Variable Models(explicit density)

VAE, Normalizing Flow



Variational Autoencoders

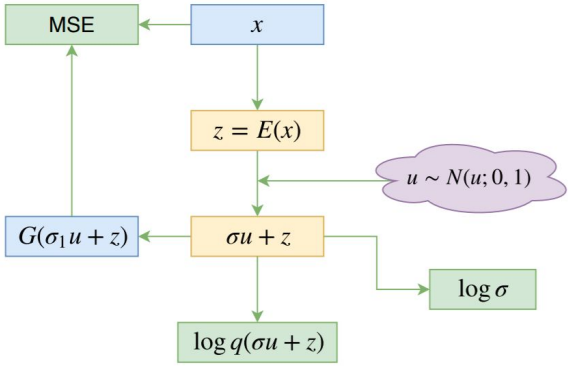
Latent variable model, and we want to learn p(z,x) where z is a latent variable, and x is our data.

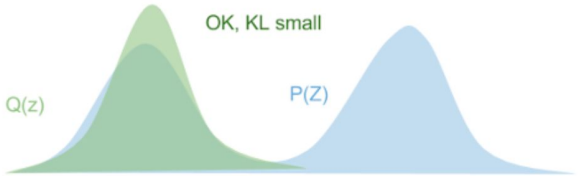
The prior p(z) is unit gaussian, and the posterior p(z|x) is approximated by a family of symmetric Gaussians (with variational inference)

Pathological Behavior - Multimodal distributions (mode matching and mode averaging)

This can potentially lead to limited expressiveness, blurry picture, posterior collapse etc..

1809.05861





Normalizing Flow - Big Trend in Science

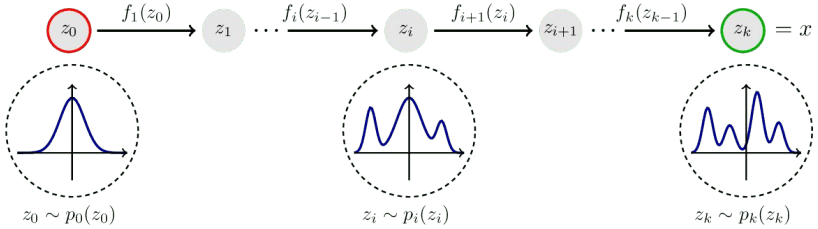
doi: 10.1126/science.aaw1147

Normalizing flow solves this problem by applying a series of invertible mappings(change of variable)

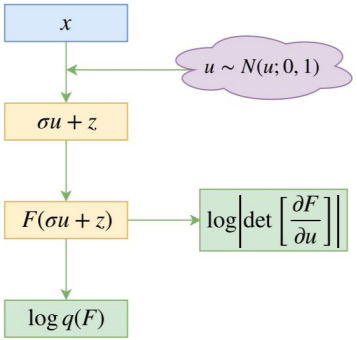
This is done via series of

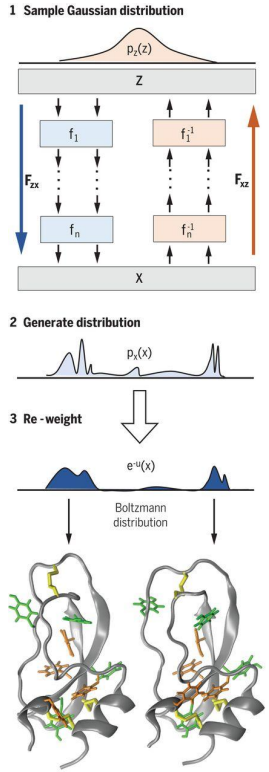
“coupling layers”, and we maximize the log likelihood directly

Popular method of modeling scientific data these days



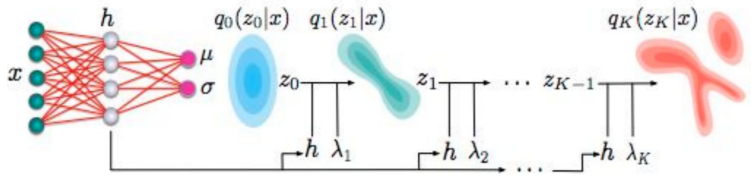
1809.05861





Normalizing flow + VAE

Lastly, we can **combine** these two ideas



Apply series of coupling layers to transform

Gaussian shape of the approximate posterior q(z|x), so that we have better approximation of the true posterior p(z|x)

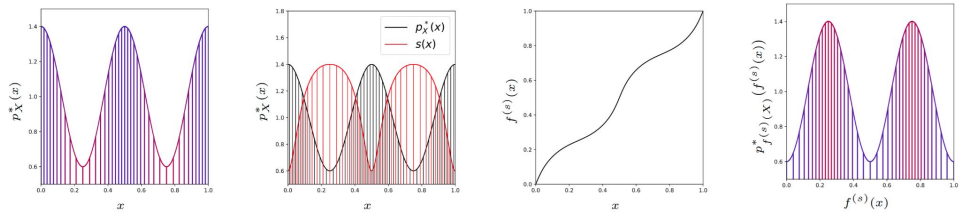
Quak uses this model, but you can use **any of density estimation method** to perform QUAK algorithm

Some Caveat

There are some pathological cases that warns us that doing anomaly detection with bare likelihood(loss) can be unideal

However, for practical purposes density estimation seems to work pretty well for anomaly detection

2012.03808



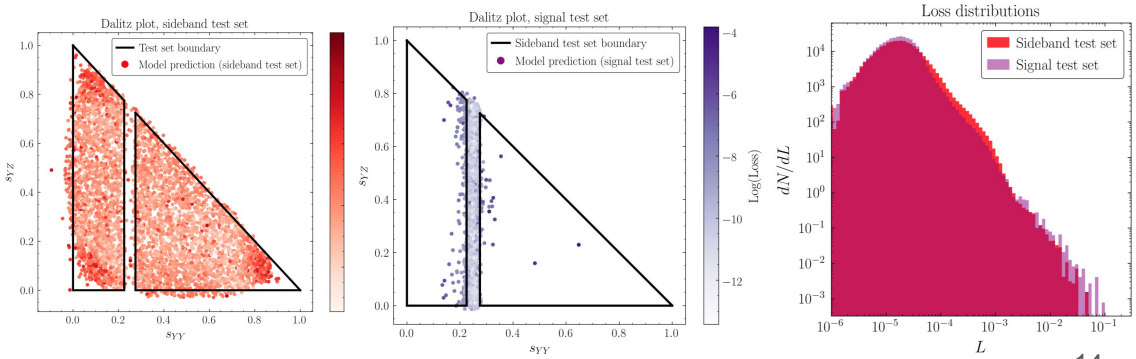
(a) An example of a distribution density p_X^*.

(b) The distribution p_X^* (in black) and the desired density scoring s (in red).

(c) A continuous invertible reparametrization $f^{(s)}$ such that $p^*_{f^{(s)}(X)}\left(f^{(s)}(x)\right) =$

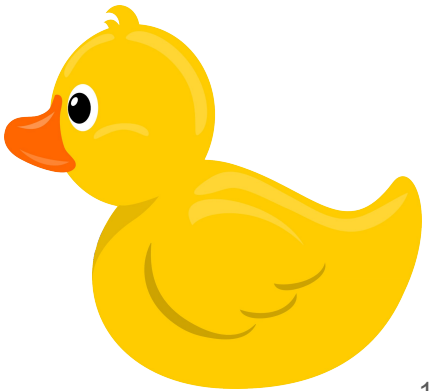
(d) Resulting density $p^*_{f^{(s)}(X)}$ from applying $f^{(s)}$ to $X \sim p_X^*$ as a function of $f^{(s)}(x)$.

2102.08380



Overview

1. Brief intro Anomaly Detection in HEP
2. ML techniques in QUAK + Technical Details
3. **Key ideas used in QUAK**
4. QUAK in Action (Signal Extraction Strategies)
5. Outlook



Goal of QUAK

Is there a way to improve sensitivity by using more information from existing signal models, **while still doing model agnostic search(preserving model independence**) ?

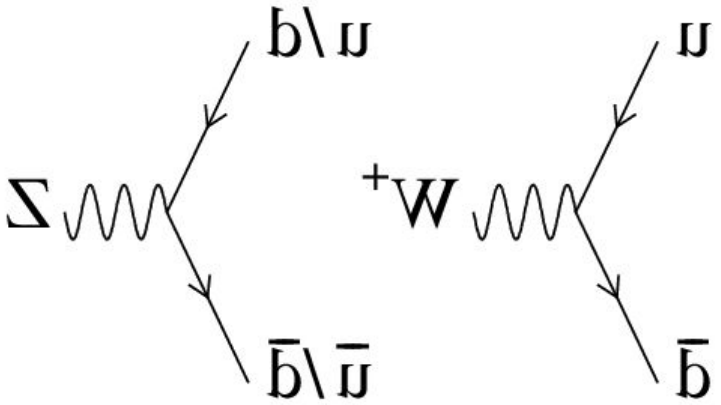
Approximate priors - We can “inject” information about the proxy that we believe should help with the search (or, some generic property we want to specifically look for)

Inspiration

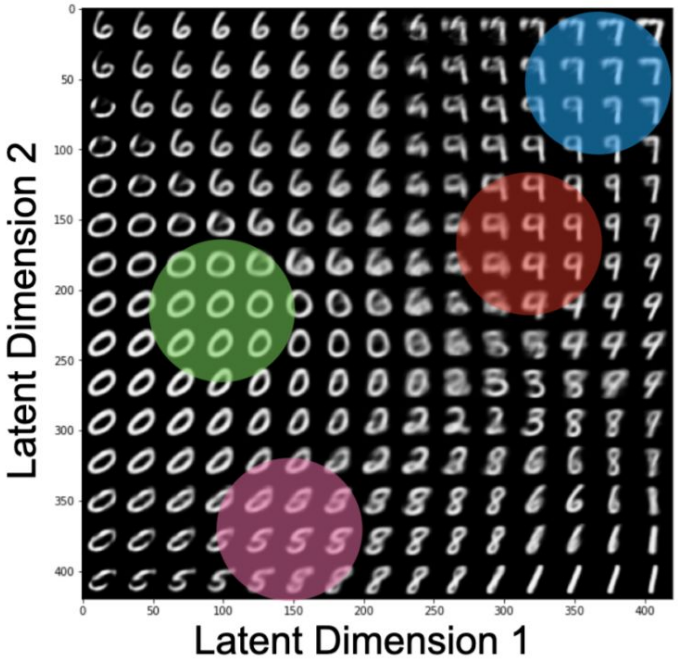
Say we discovered W boson, but not Z yet

Different mass, charge, etc but they share some properties, such as two prong hadronic decay

Understanding data distribution from W decay would help with discovery of Z



Toy Scenario



MNIST Digits dataset

Lets try to detect digit 9(anomaly) buried in dominant background, digit 5

Doesn't make sense to use only the information from 5 if

we know other digits (7 is quite similar to 9)

We are building intuition by visualizing a metric space- space is the latent space, distance is the euclidean distance in the latent space

We actually used this as playground to test the ideas

We will see the actual result of the experiment later

QUAK in different settings

QUAK can be tried and tested in many different settings

- **CMS Open Data**: Say that we've discovered W boson and know QCD well, can we discover Z boson event? → W boson would be an approximate prior
- **MNIST** : If we have abundant number 5 samples as our background, try to detect a new number 9(actual signal) → 7 can be an approximate prior (including other numbers work as well)
- **LHC Olympics**: Given the QCD events, can we discover embedded BSM physics event in the black boxes? → choose any one BSM model as our approximate prior
- Higgs physics : By studying well known decay modes of the Higgs, can we improve not so well studied decay modes?

Can we use information from what we already know?

In dijet searches, with density estimation based approaches you train a model to learn the distribution of background (QCD) and choose events with low p(background)

Unsupervised searches use only information of background (SM physics)

Can we incorporate certain aspects of known physics into the search?

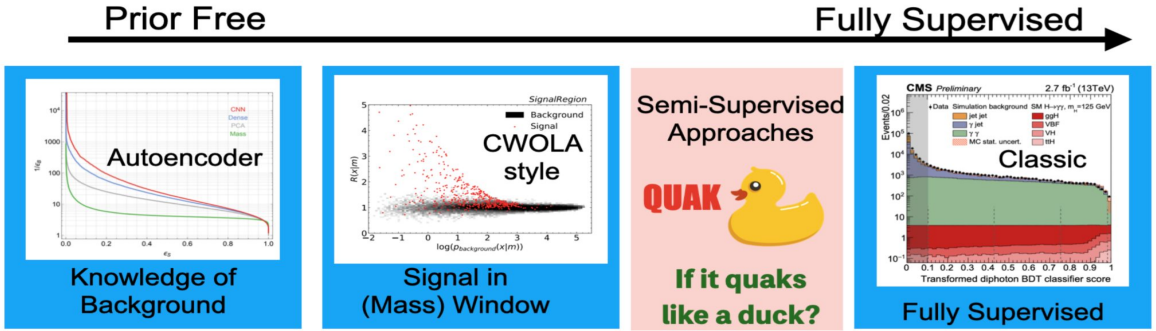
Incorporating signal models into the search can improve the sensitivity

Semi-supervised methods try to do this (Many ways to do this!)

Choice of loss metric, training on ensemble, and many more

Key is to **preserve model independence** while incorporating some aspect of signal data

2011.03550

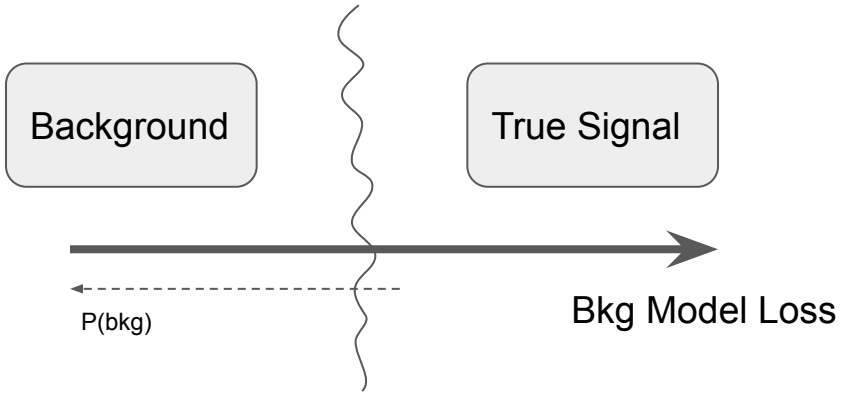


One way to do this - QUAK

Quasi Anomalous Knowledge: Searching for new physics with embedded knowledge
2011.03550

Conventional density estimation based search

learning latent representation of background (train on background)



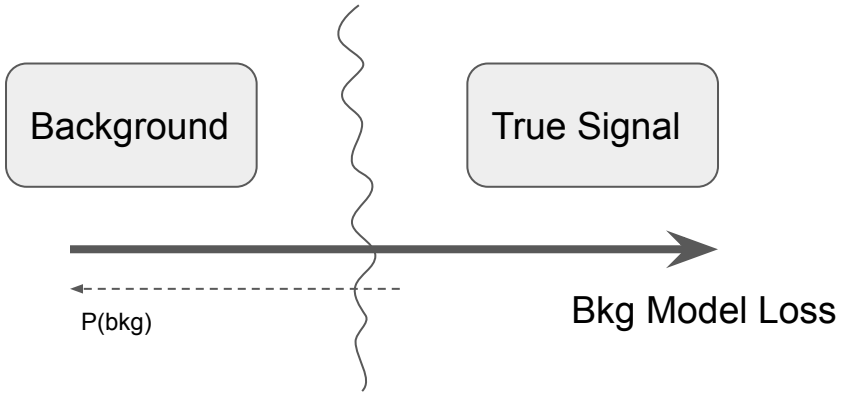
Loss: MSE reconstruction loss $\left|\left|x - x'\right|\right|$
Which we are using as proxy for Negative Log likelihood

One way to do this - QUAK

Quasi Anomalous Knowledge: Searching for new physics with embedded knowledge 2011.03550

Conventional density estimation based search

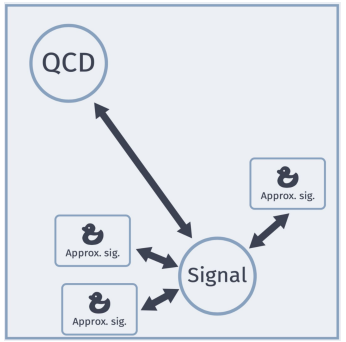
learning latent representation of background (train on background)



Now, we introduce
“approximate signal prior”
: It is a dataset we train another density estimation algorithm on

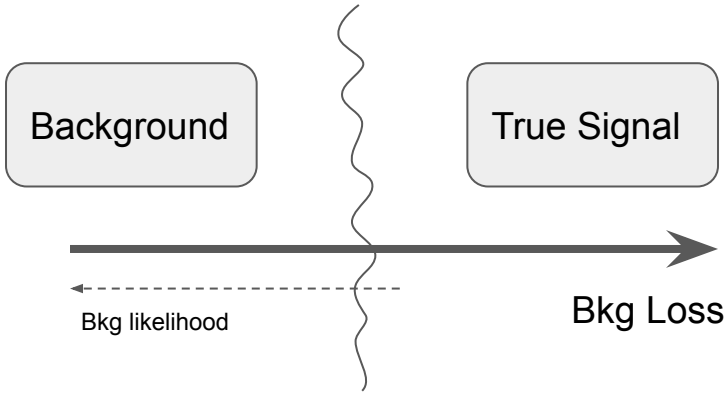
Lets call loss from this model as “signal loss”, and see it as a proxy for p_{sig}

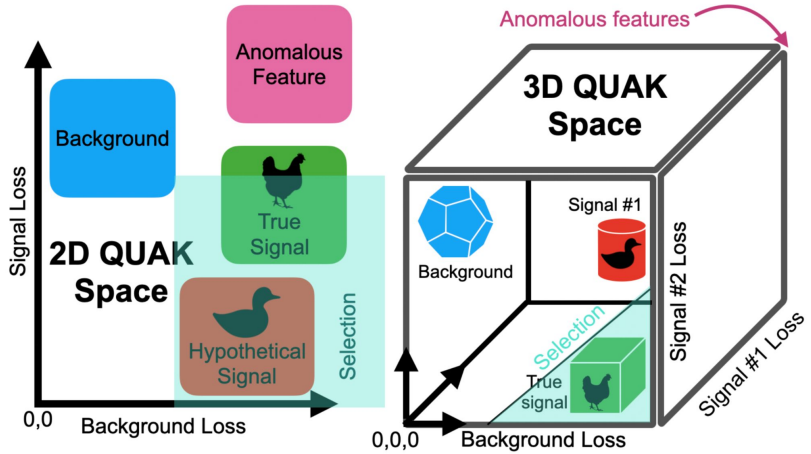
Core Idea of QUAK



QUAK adds another axis to this search by training multiple models on approximate hypothetical signal priors

This gives us control we didn't have in 1D,can separate out more categories





2011.03550

We also checked model independence and performed comparison with supervised methods

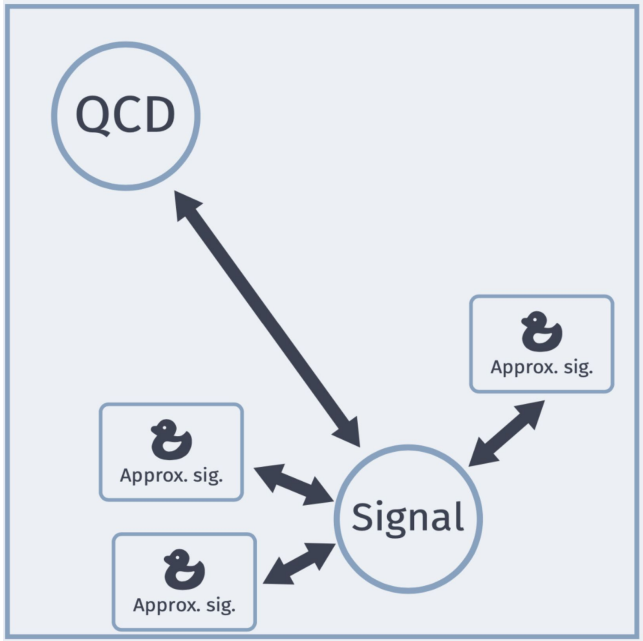
From metric space perspective

Say we have a **robust distance metric** such as good approximation of likelihood through generative modeling (or OT based metric(p-Wasserstein))

What we do(in context of QUAK):

We train new generative model for each signal priors, Run the same training multiple times

Each event evaluated with multiple distance(in our case, likelihood)



QUAK Algorithm

1. Train a model on dominant background,
2. Introduce N-1 approximate priors where we train our model on
3. Build N dimensional space of loss for a dataset
4. Scan the space to search for anomalies and extract different features

By doing this, we

1. Build a more complex space by adding dimension
2. In some space with a distance metric, we add more reference points to calculate distance with respect to

Side note : Strategies that see no signal information

“Signal Information” : Information from non-background data, Such as data generated with a specific BSM model

There are methods that use **no signal information** at all

- **Either rely purely on data** (overdensity detection) or use **only the background** information

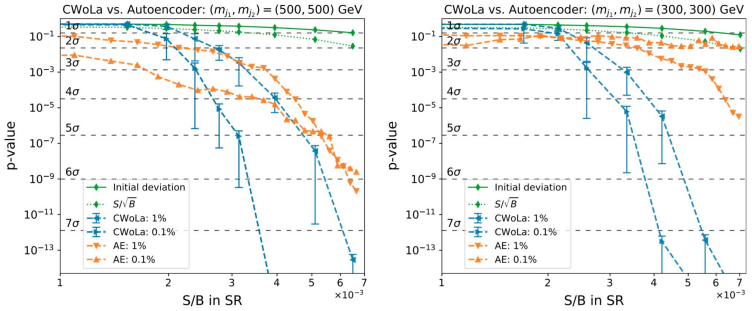
ex) CWoLa (1708.02949)

Uses only data - try to detect overdensity

ex) Pure autoencoder based (2110.08508)

Training one autoencoder on the dominant background

2104.02092



Different ways of incorporating physics knowledge

1. Train a supervised network

Train (A vs B classifier, and apply it to A” vs B task)

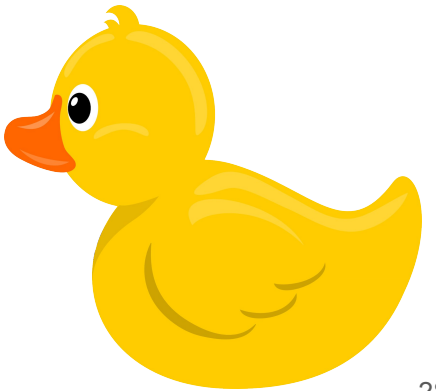
Does not preserve model independence

Explanation : High dimensional decision boundary

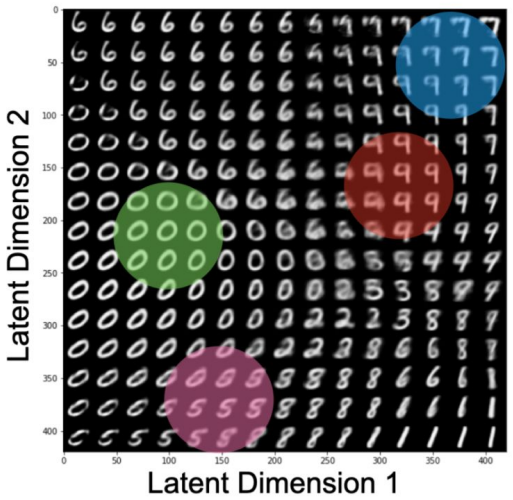
2. Semi - supervised networks with mixture training/ modified loss (2007.01850)

Overview

1. Brief intro Anomaly Detection in HEP
2. ML techniques in QUAK + Technical Details
3. Key ideas used in QUAK
4. **QUAK in Action (Signal Extraction Strategies)**
5. Outlook



MNIST Experiment



We have the handwritten dataset:

Here the anomaly we try to detect is digit 9

Our dominant background is the digit 5

However, we will use other digits as our **approximate priors**

With this context, we performed various experiments to decide **what is the best way** to incorporate signal priors

Details about the studies

Dataset

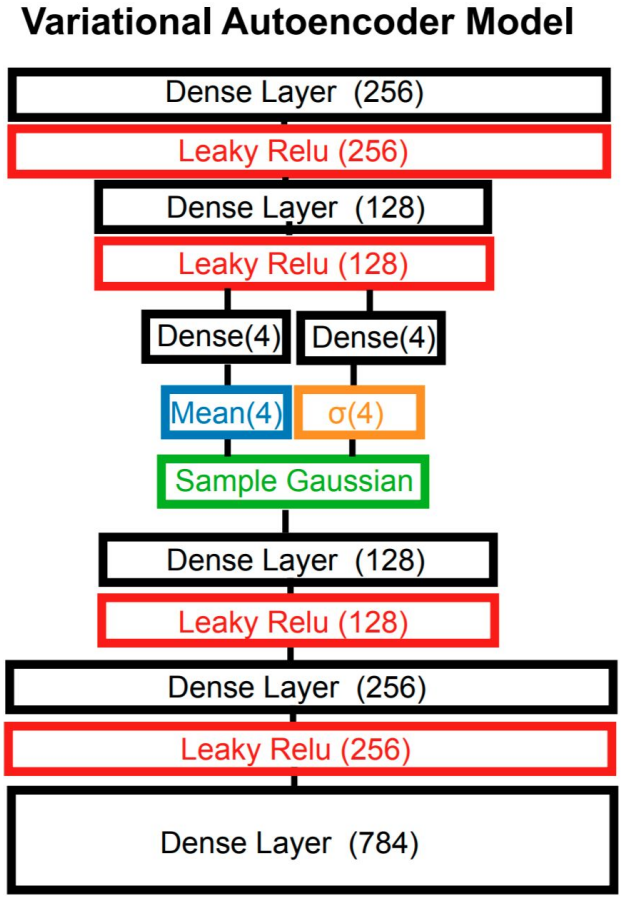
26 pixels by 26 MNIST Images

NN architecture

Encoder, Decoder are MLPs,3 dense layers on either end and variational inference, four dimensional Gaussian latent encoding.

Loss

MSE reconstruction loss on the 784 dim images + KL divergence between p(z) and q(z|x)



Choices we can make

Choice of **how to incorporate** “approximate” signal priors

1. Train one model that sees all the priors
2. Train multiple models that learns the distribution of different data

Choice of **space**

1. Loss space vs Latent space

QUAK is quite flexible; What would give the best anomaly detection result?

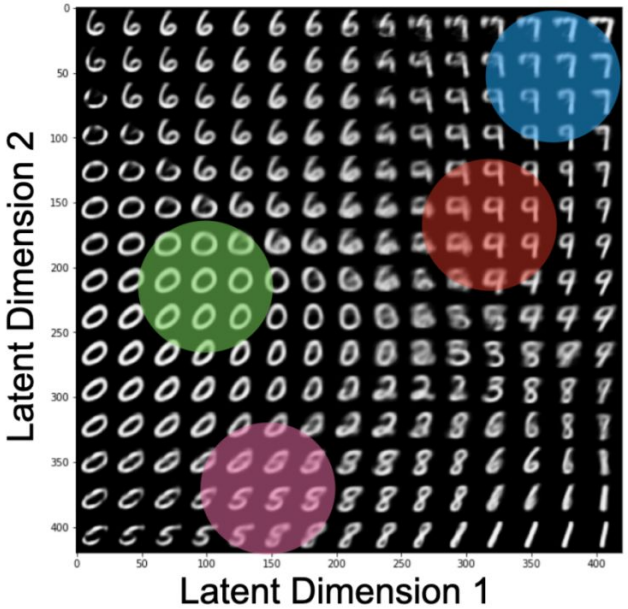
Flexibility of choosing approximate prior

We know intuitively and from latent space diagram that 7 is a pretty good choice of proxy for 9

Looking at digit 0 in the latent space, we see its equidistant from 5 and 9

Lets dilute our approximate prior dataset of pure 7 by mixing in different proportions of 0

If the performance doesn't degrade much, we see the choice of approximate prior doesn't have to be very accurate



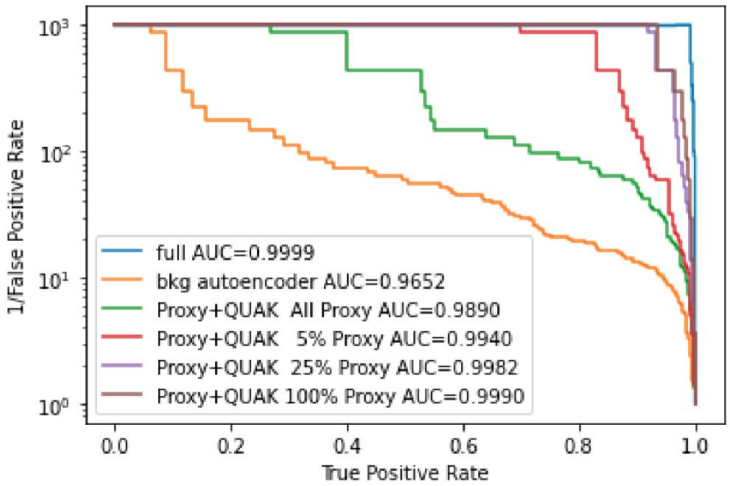
Flexibility of choosing approximate prior

Compare the red line and the orange line

Just with 5% of 7's (95% of 0's) in the approximate prior, we see significant boost in performance

Another point is comparing the green and the orange line

Even though 0 is a pretty bad approximate prior, it still helps a lot with the search!



Latent vs Loss space

Another speculation : Can we do anomaly detection in the latent space, with encoded latent variables?

In this case you would train a **single VAE** with N (background+signal) priors and do anomaly detection in the latent dimension z

How would this compare to QUAK, where we try anomaly detection in loss space?(as a proxy for likelihood for each prior)

In this case **N separate models** will be trained

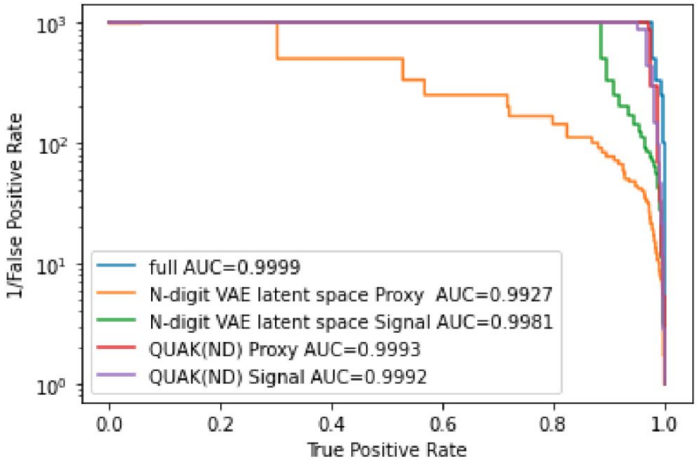
Latent vs Loss space

1. By comparing (orange+green) and (red+purple) we see that QUAK generally outperforms latent space method

Proxy, and signal means how we calculate the anomaly score

Train a supervised NN (bkg vs proxy(or signal)) in high dimensional latent/loss space to get anomaly score

Orange, Red are realistic scenarios



2. QUAK is **more robust** to using proxy for getting anomaly score(doesn't move around much)

Orange - Green large shift, purple-red small shift

Our conclusion

QUAK can be **extended** in **several directions**

Depending on the structure of data the optimal algorithm will change

Choice of what approximate signal prior to add / subtract and the way we construct and scan QUAK space will bias the search in certain direction

While adding approximate signal priors help improve sensitivity, it still preserves model independence.

Approximate signal priors doesn’t have to be accurate to help with the search

LHC Olympics Dataset

Dijet anomaly detection challenge

Start with two datasets

- **Dataset of QCD : 1M background events**
- **Black Box dataset: 1M events with unknown number of anomalous events mixed in**

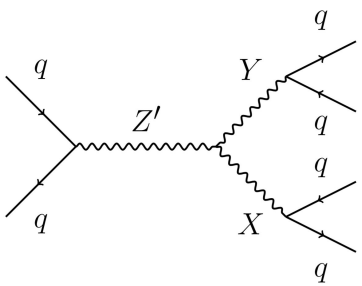
Dataset Features

120 Particles from di-jet events, 3 features (pT, eta, phi) per particle

QUAK Strategy

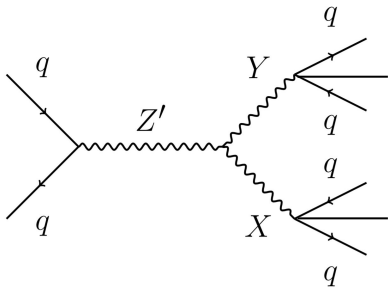
In order to add approximate signal hypothesis and to perform studies we generated various dijet signal hypothesis, with varying pronginess and masses

Approximate Signal Prior



2 prong - 2 prong

$M_{Z'}$, M_X, M_Y Varied



3 prong - 3 prong

$M_{Z'}$, $M_X = M_Y$ Varied

Details about the studies

Dataset

120 Particles from di-jet events, 3 features (pT, eta, phi) per particle

Preprocessing

Cluster particles to jets, save top 2 highest pt jets, summarize each evt to 12 jet/substructure variables (6 variables per jet)

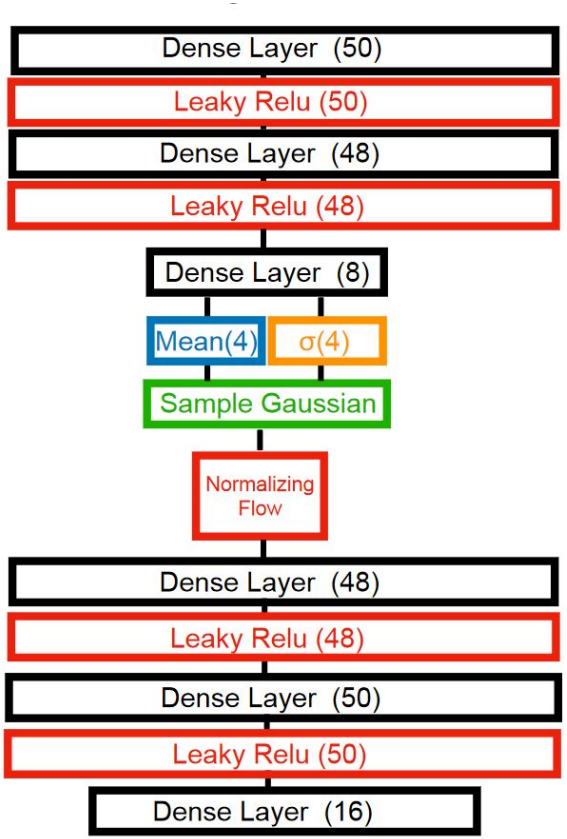
$$m_J, \sqrt{\tau_1^{(2)}}/\tau_1^{(1)}, \tau_{21}, \tau_{32}, \tau_{43}, n_{\mathrm{trk}},$$

NN architecture

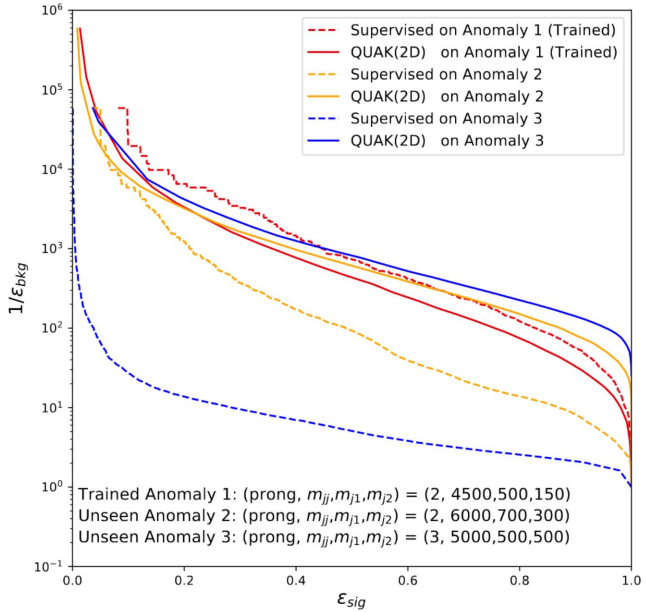
Encoder, Decoder are MLPs, and variational inference, normalizing flow transform (MAF:SOTA) simple gaussian shape of the prior.

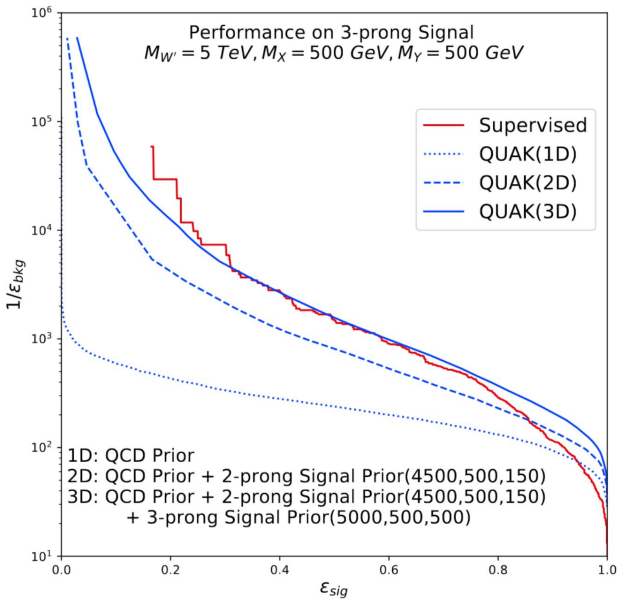
Loss

MSE reconstruction loss on 12 features + KL divergence between p(z) and q(z|x), with beta scaling



What we show in the paper





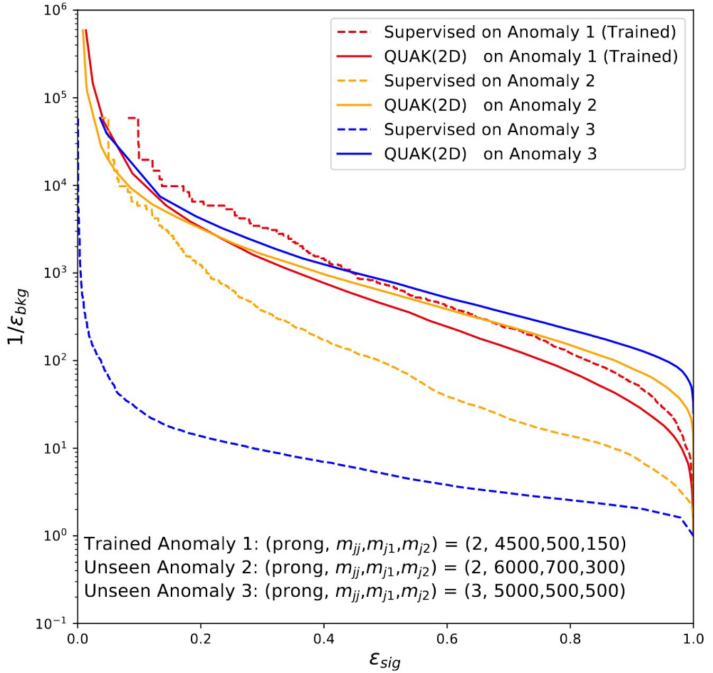
Model Independence

Most important question is whether it preserves the **model independence**

We compare 2D QUAK and Supervised Network(different ways of incorporating prior)

2D QUAK : Built from two models trained on QCD, and Anomaly 1

Supervised: A model trained to do QCD vs Anomaly 1 (Targeted search)

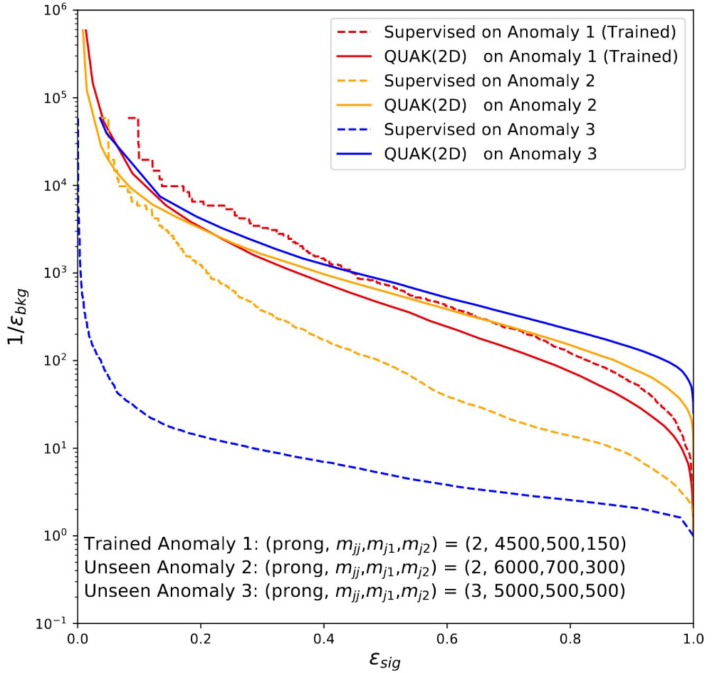


Model Independence

2D QUAK : Built from two models trained on QCD, and Anomaly 1 ——————————

Supervised: A model trained to do QCD vs Anomaly 1 (Targeted search) - - - - - - - - - - -

Compare two **red lines**: These are the cases when we injected signal prior that exactly matches anomaly we were looking for

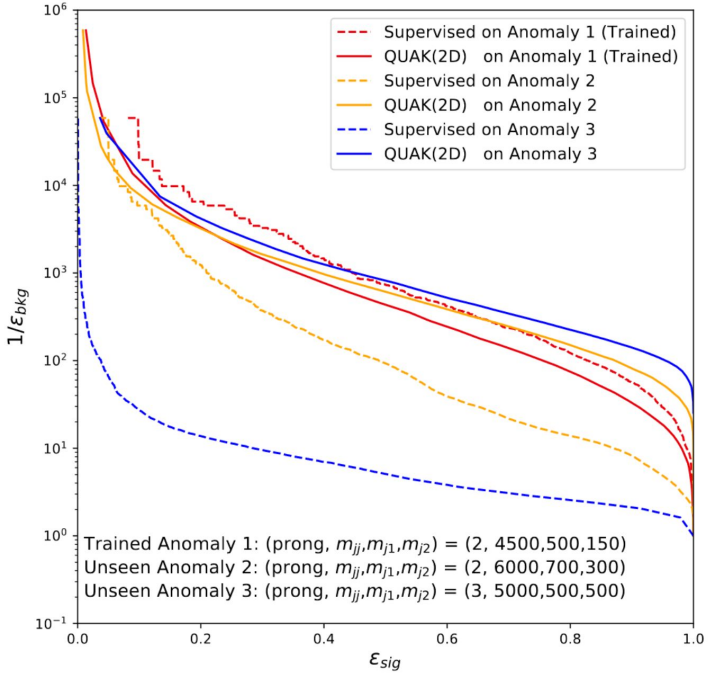


Model Independence

2D QUAK : Built from two models trained on QCD, and Anomaly 1 ——————————

Supervised: A model trained to do QCD vs Anomaly 1 (Targeted search) - - - - - - - - - - -

Compare two **yellow lines**: These are the cases when we injected signal prior that is a bit different from anomaly we were looking for



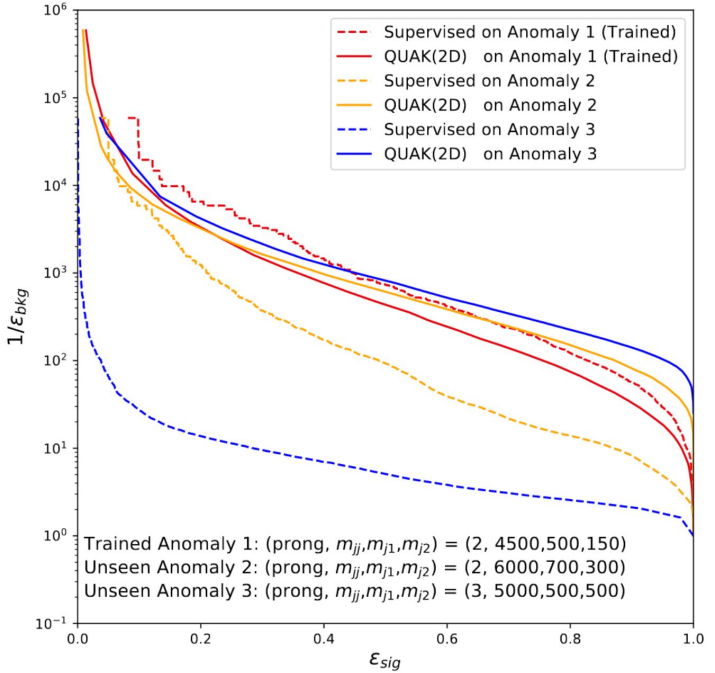
Model Independence

2D QUAK : Built from two models trained on QCD, and Anomaly 1 ——————————

Supervised: A model trained to do QCD vs Anomaly 1 (Targeted search) - - - - - - - - - - -

Compare two **blue lines**: These are the cases when we injected signal prior that is quite far from anomaly we were looking for

supervised performance breaks down when the target is far from the training

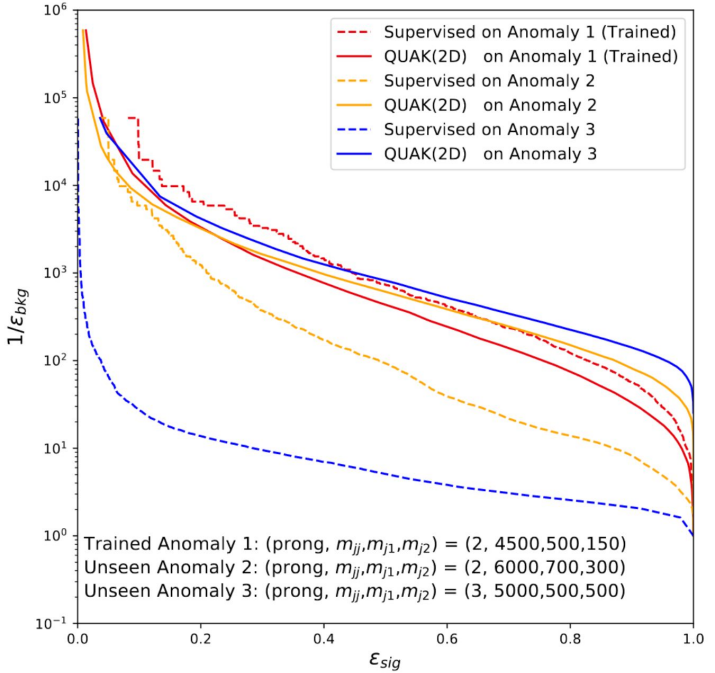


Model Independence

1. Compare all solid lines, they are grouped together, unlike dashed lines

   Approximate signal priors don't have to be too accurate to boost the search performance significantly

2. QUAK performance is very stable unlike supervised performance

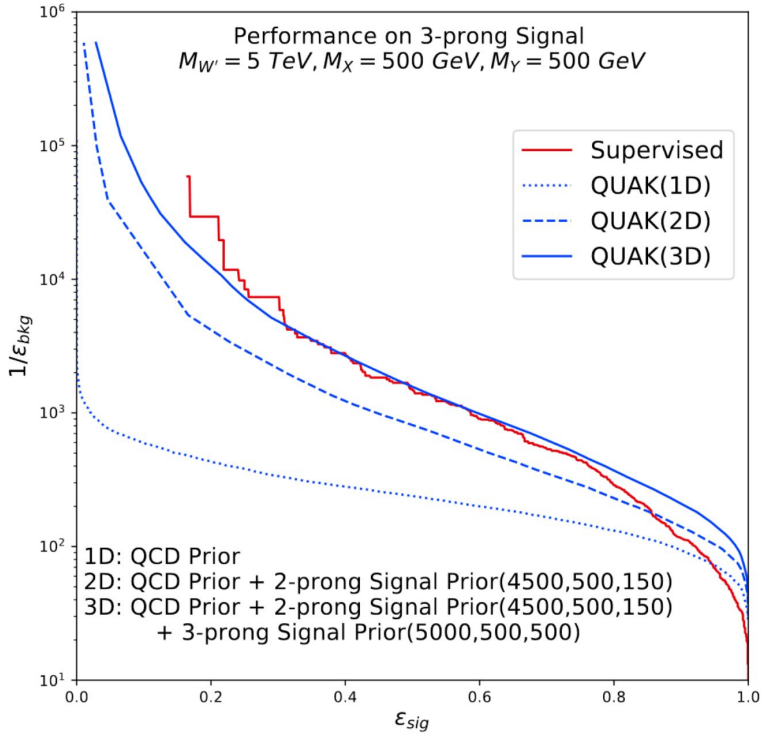


What we show in the paper

Compare three blue lines:

As we add more priors, the performance gets better

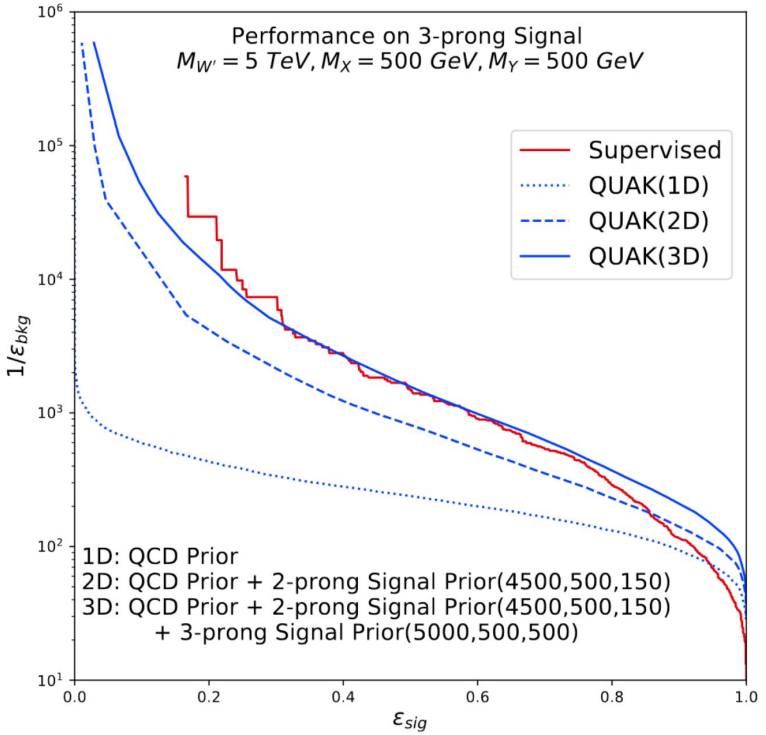
We observe that as soon as we include a prior that accurately describes the anomaly we are looking for, the performance is comparable to the supervised line



What we show in the paper

1. We can **approach supervised classifier's performance** if we have accurate prior

2. We again observe that the **approximate signal prior doesn't have to be too accurate** to significantly boost the performance of the search



Signal Extraction Strategy - Black Box

We applied quak to black box 1 dataset

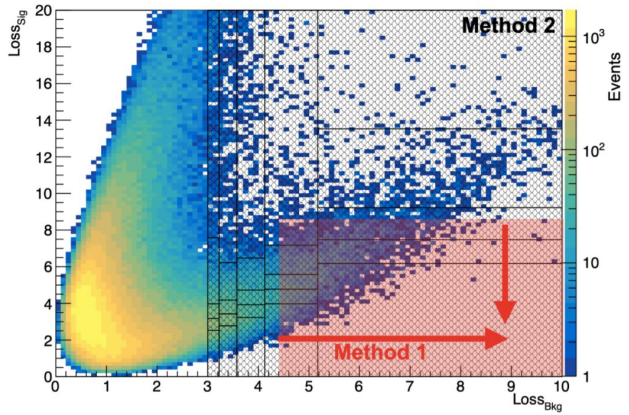
One of the points we spent most time in

Straightforward method - Brute force scan

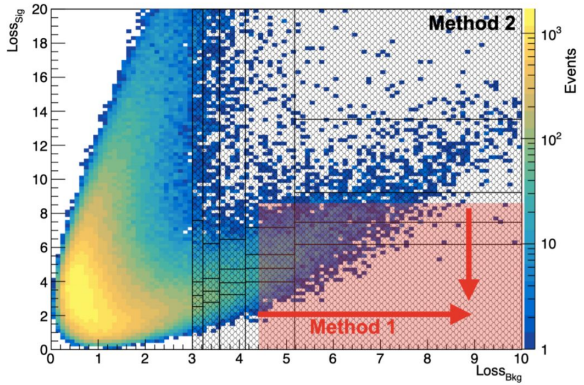
Becomes expensive as dimension of the space increases.

In MNIST case, we could train a classifier with a proxy (Not always the case)

In the end we came up with two methods.

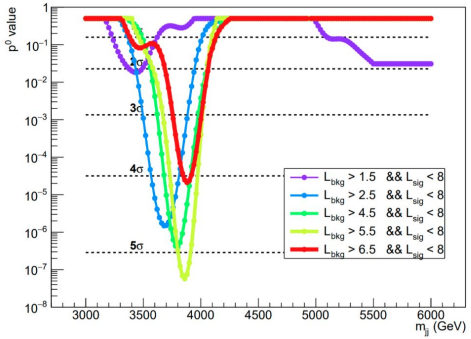


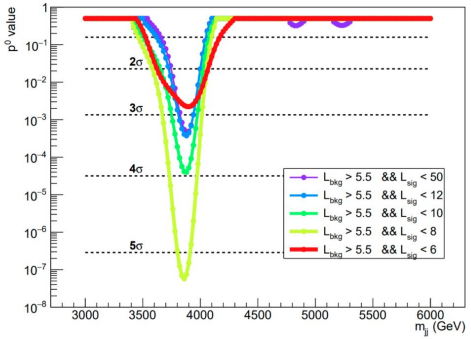
Signal Extraction Strategy - Method 1

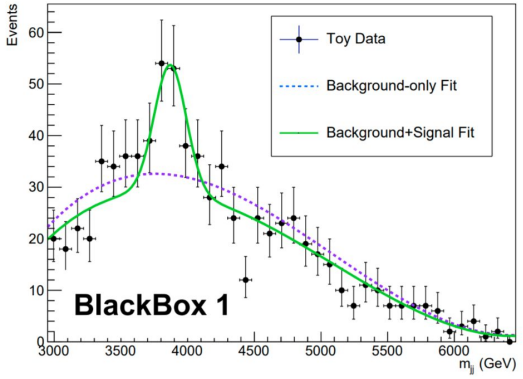


X axis

Y axis







Full 2d scan of the loss space

Fit background, signal with a fixed functional form

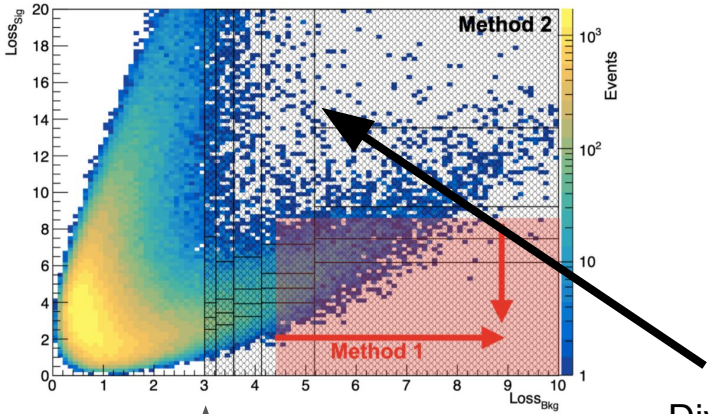
(Bernstein Poly, Gaussian)

Good way to find an excess

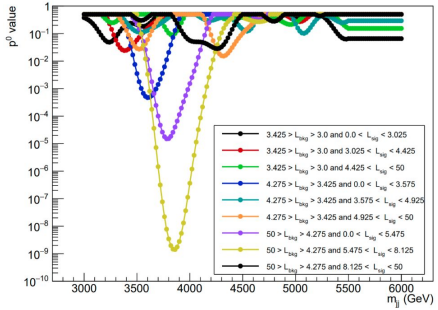
Signal Extraction Strategy - Method 2

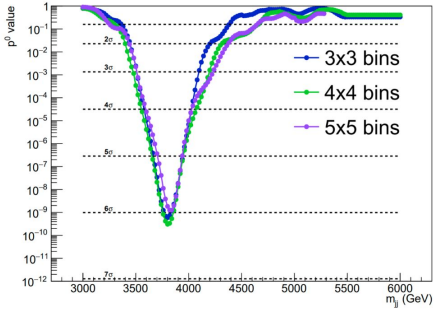
We use slightly different strategy for quoting a significance of the excess

Method 1 is biased towards finding a large excess, whether its real or not(LEE)



Selection at ~ 2% background probability



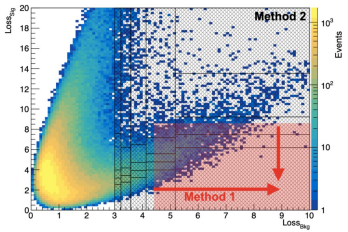


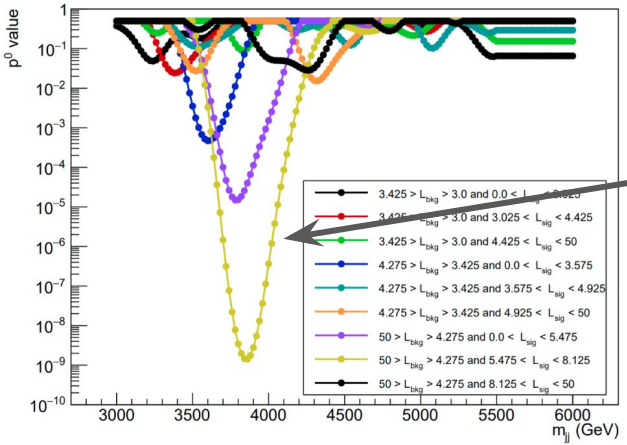
Divide into 2d bins that have about equal number of events (divide along background direction first)

Signal Extraction Strategy - Method 2

We use slightly different strategy for quoting a significance of the excess

Method 1 is biased towards finding a large excess, whether its real or not



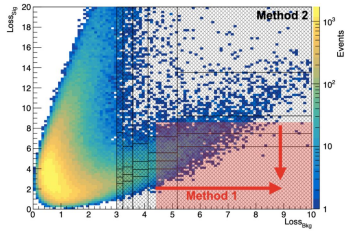


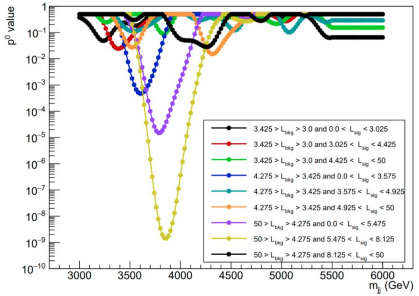
One bin dominates the significance: Signals go to a **specific region** of the QUAK space: We have locality

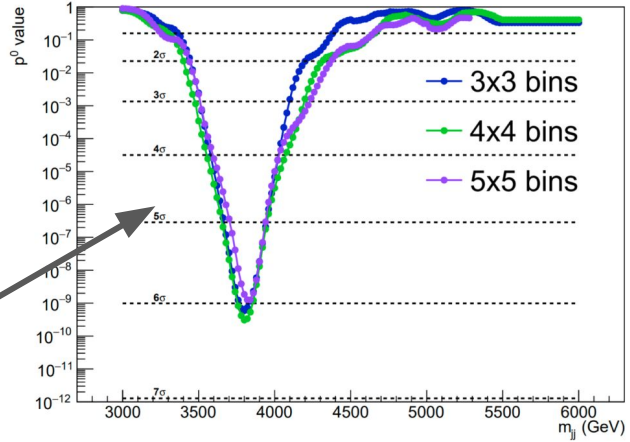
Signal Extraction Strategy - Method 2

We use slightly different strategy for quoting a significance of the excess

Method 1 is biased towards finding a large excess, whether its real or not







Treating each bin as independent experiment, we can combine the p value of each bin to quote as our final p value (Combining p-values make the search robust against LEE)
Check the **LEE** with a dataset with no injected signal, and maximum fluctuation p value was **less than 2 sigma**

Black Box 1 results

Signal: 834 events / 1M

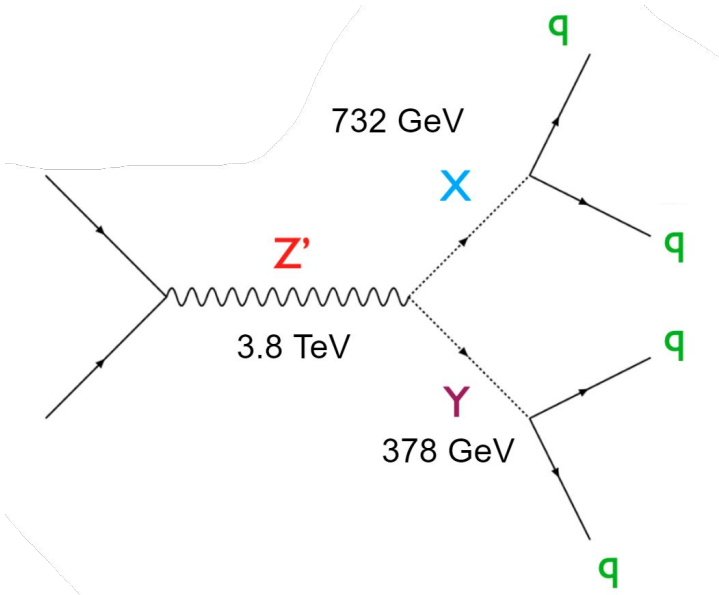
Z'->XY; X,Y->qq

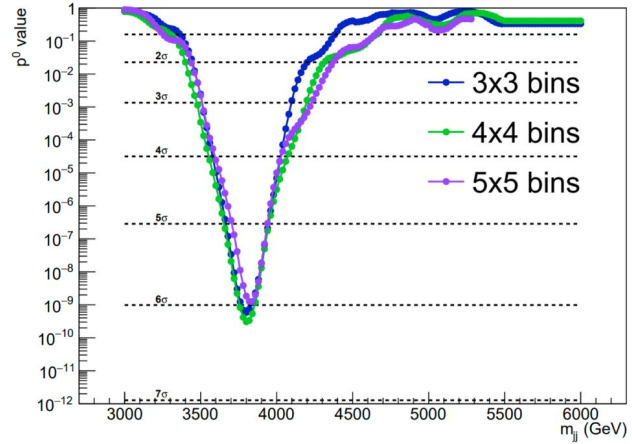
mZ' = 3823 GeV

mX = 732 GeV

mY = 378 GeV

Quak method robustly finds hidden signal





Our Conclusion

QUAK allows us to do **model agnostic search while improving sensitivity** over wide variety of signal

QUAK allows us to inject knowledge of physics into the search through “approximate priors” by incorporating some generic feature that we believe the new physics should have

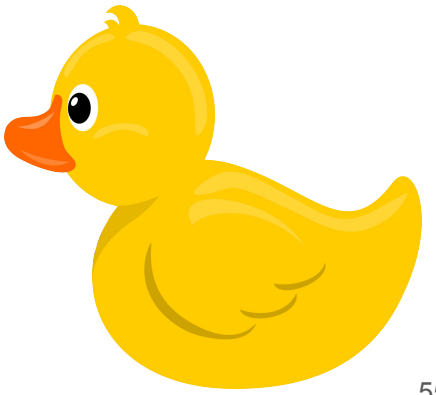
These approximate priors **don’t have to be accurate** to improve sensitivity

We can always be more ambitious and build a larger QUAK space to improve sensitivity

And if we use correct prior, it can reach/exceed performance of a supervised classifier

Overview

1. Brief intro Anomaly Detection in HEP
2. ML techniques in QUAK + Technical Details
3. Key ideas used in QUAK
4. QUAK in Action (Signal Extraction Strategies)
5. **Outlook**



Outlook

QUAK can be improved / developed in many directions

- How to efficiently scan the N-dim QUAK space

QUAK can also be applied in many different settings

- Only tested in dijet anomaly setting, but there can be many other applications
- ex) Invisible decay of Higgs

It could be put in trigger!

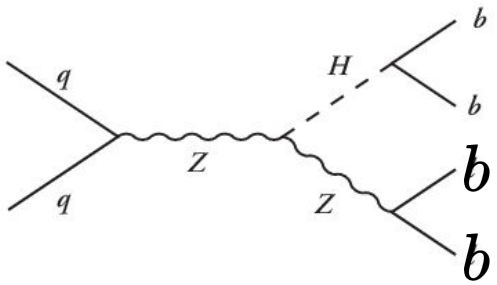
QUAK in physics analysis

QUAK allows us to do many searches at once!

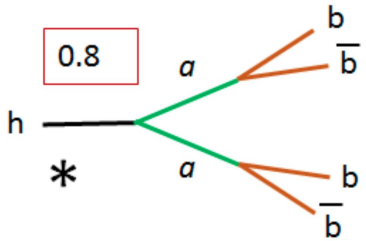
Ex) QUAK will allow us to do search that is sensitive to all dijet anomaly events, by just injecting one prototypical dijet anomaly event

Regardless of (prong, mjj, mj1, mj2) target

Another fun idea)



SM process(Approximate prior)



BSM scalar(Anomaly)

Can we run QUAK online?

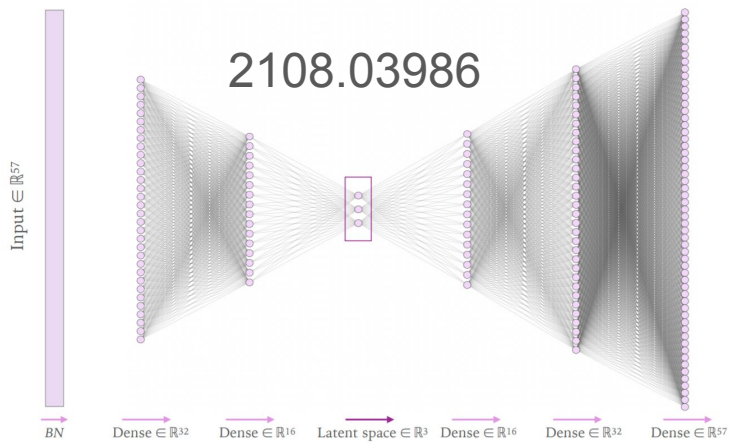
Algorithms should meet these two criteria to be run online:

1. Only look at data once
2. Be able to meet throughput and latency constraints

QUAK satisfies both criteria

(Autoencoder based methods are good candidates to be used in online setting)

Hardware acceleration: deep autoencoder can be put on FPGAs, meeting L1 trigger constraints (2108.03986)



Thank you!