Training toward significance with the decorrelated event classifier transformer neural network

Jaebak Kim[*]

Department of Physics, University of California, Santa Barbara, California, USA

(Received 18 January 2024; accepted 24 April 2024; published 24 May 2024)

Experimental particle physics uses machine learning for many tasks, where one application is to classify signal and background events. This classification can be used to bin an analysis region to enhance the expected significance for a mass resonance search. In natural language processing, one of the leading neural network architectures is the transformer. In this work, an event classifier transformer is proposed to bin an analysis region, in which the network is trained with special techniques. The techniques developed here can enhance the significance and reduce the correlation between the network's output and the reconstructed mass. It is found that this trained network can perform better than boosted decision trees and feed-forward networks.

DOI: 10.1103/PhysRevD.109.096035

I. INTRODUCTION

Experimental particle physics is often performed by colliding particles and observing their interaction with detectors. Particles generated from the collisions are investigated by reconstructing the detector data. A common method used in particle searches is to search for a resonance in the reconstructed mass distribution. An example of this resonance is shown in Fig. 1. Signal events create a peak that can be seen above background events. To estimate the number of signal and background events, the mass distribution is fitted with a peaking signal and a smooth background function. The fitted mass region is set to be larger than the signal peak width to estimate the background more precisely. This method is referred to as "bump hunting" and was used to search for the Higgs boson [1,2]. The sensitivity of the method can be quantified in terms of significance [3,4]. A larger significance corresponds to a smaller probability that the bump was created by random statistical fluctuations of the background.

To increase the expected significance of the bump hunting method, requirements are applied to create a search region to suppress the background, while preserving the signal. The sensitivity of the analysis can be further enhanced by "binning," where the search region is divided into multiple bins according to other variables. However, if the binning affects the reconstructed mass distribution of the background, estimating the number of signal events can become difficult. An extreme case would be one where the background peaks similarly to the signal. Therefore, a desirable characteristic of binning in the other variables is that it does not affect the shape of the background distribution. Binning can be performed using distinguishing features of the signal [2], or by using machine-learning techniques [1,5,6] that classify the signal and background events. A common machine-learning technique used for this task is the boosted decision tree (BDT).

II. EVENT CLASSIFIER TRANSFORMER NEURAL NETWORK

Transformer neural networks [7] are the leading neural network architecture in the natural language processing field. The architecture has been applied to particle physics

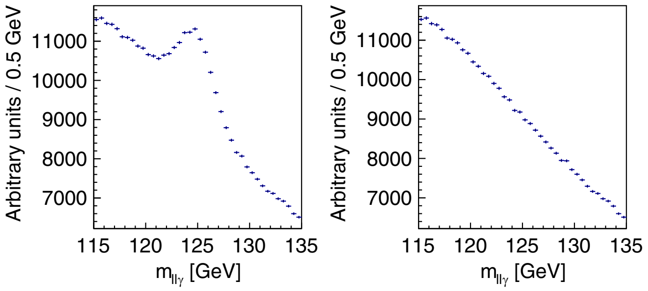

FIG. 1. Left: mass of reconstructed Higgs boson candidates from the $H \to Z(\ell^+\ell^-)\gamma$ decay, where a bump can be seen due to the presence of the Higgs boson particle. The Higgs boson cross section was scaled up by 100 to make the bump visible. Right: mass of reconstructed Higgs boson candidates from the $H \to Z(\ell^+\ell^-)\gamma$ decay with the nominal Higgs boson cross section, where the bump cannot be seen due to the background.

[*] jbkim@charm.physics.ucsb.edu

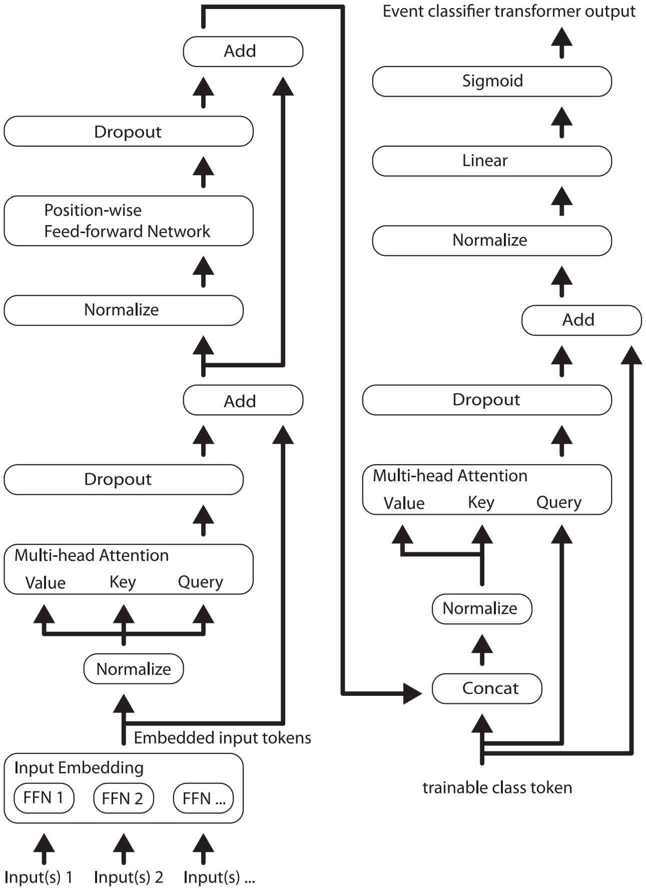

FIG. 2. Architecture of the event classifier transformer. “FFN” refers to a feed-forward neural network. “Normalize” refers to layer normalization. “Add” refers to the implementation of the residual connection. “Concat” refers to a layer concatenating tokens. “Linear” refers to a linear layer.

in a network called Particle Transformer [8]. The network identifies the type of particle that produced a jet, which is a cluster of hadrons whose momenta lie within a cone. In this work, a new transformer-based neural network called an event classifier transformer is proposed that classifies signal and background events to bin a bump hunt analysis.

The architecture of the event classifier transformer is shown in Fig. 2. The inputs are the features of the event, such as the momenta or angles of particles. To be able to use the transformer architecture, each input or each set of inputs is “embedded” into a token, which is a set of N_{repr} numbers, using separate feed-forward neural networks. In contrast to the transformer, there is no positional encoding, because event features do not have sequential dependence. The tokens are normalized using layer normalization [9] and passed to a multiheaded attention layer (MHA) [7]. The output is summed to employ a residual connection [10] and normalized with layer normalization. A positionwise feed-forward neural network [7] is applied with residual connection and layer normalization, where the same positionwise network is applied to each token. The output is a group of tokens, where each token is a contextual representation of the input feature. Following a simplified version of the CaiT [11] approach and the Particle Transformer, a trainable class token is passed as a query to a MHA, and the contextualized tokens concatenated with the class token are passed as a key and value. The output employs a residual connection and layer normalization and is passed to a linear layer. The linear layer output is a single value that is passed through a sigmoid to represent the probability that the event is from the signal process. To prevent overtraining, dropout layers [12] are used in the scores of the MHA and before the residual connection during training. Implementation of the network is publicly available at Ref. [13].

III. TRAINING TECHNIQUES FOR ENHANCING SIGNIFICANCE

Special training techniques are developed to apply to the event classifier transformer and other neural networks, to increase the expected significance and reduce the correlation for a bump hunt analysis. The following new training techniques are investigated:

(1) Specialized loss function with mass decorrelation.
(2) Data scope training.
(3) Significance-based model selection.

These techniques are described in turn below, and implementation is publicly available at Ref. [13].

A. Specialized loss function with mass decorrelation

In a binary classification task, where the input x is provided to predict the label y, a neural network $f(x)$ can be trained to output a prediction $\hat{y} = f(x)$. In this work, signal is defined to be $y = 1$ and background is defined to be $y = 0$. The network is trained with a loss function that is used to minimize the difference between the network's output $\hat{y}$ and label y.

A common loss function is the binary cross-entropy (BCE) loss [14],

$$\mathrm{BCE}(\hat{y}, y) = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y}), \tag{1}$$

where the minimum corresponds to the condition

$$\frac{\partial \mathrm{BCE}}{\partial \hat{y}} = 0 \tag{2}$$

and is achieved when

$$\frac{\hat{y} - y}{\hat{y}(1 - \hat{y})} = 0, \tag{3}$$

which implies $\hat{y} = y$.

In this work, to increase the penalty when $\hat{y}$ and y are different, while keeping the property that the minimum is

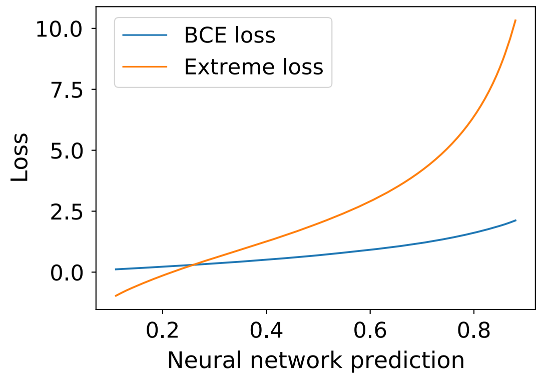

FIG. 3. BCE loss vs extreme loss, when the label is $y = 0$. Extreme loss penalizes the neural network more than BCE loss for network predictions that are close to 1.

achieved at $\hat{y} = y$, an alternative loss inspired from BCE called extreme loss, $E(\hat{y}, y)$, is proposed:

$$\begin{aligned} E(\hat{y}, y) &= \int \frac{\hat{y} - y}{\hat{y}^2(\hat{y} - 1)^2} d\hat{y} \\ &= -y\left[-\frac{1}{\hat{y}} - \ln(1 - \hat{y}) + \ln(\hat{y})\right] \\ &\quad - (1 - y)\left[\frac{1}{\hat{y} - 1} + \ln(1 - \hat{y}) - \ln(\hat{y})\right]. \end{aligned} \tag{4}$$

For a given value of $\hat{y} - y$, the extreme loss penalizes the neural network more than the BCE loss, as can be seen in Fig. 3. This extreme loss heavily suppresses backgrounds that have network predictions close to 1, and also signals that have network predictions close to 0. Because the expected significance when binning a bump hunt analysis is sensitive to backgrounds that have high network predictions, this loss can help to increase the significance.

To decorrelate the neural network output with the reconstructed mass, distance correlation (DisCo) regularization is used [15]. DisCo measures the dependence between $\hat{y}$ and mass, where the value of DisCo is zero if and only if $\hat{y}$ and mass are independent. DisCo is multiplied by a factor λ and added to the classifier loss function:

$$\mathrm{Loss} = \mathrm{Loss}_{\mathrm{classifier}}(\hat{y}, y) + \lambda \cdot \mathrm{DisCo}(\mathrm{mass}, \hat{y}). \tag{5}$$

The DisCo term penalizes the neural network when $\hat{y}$ and mass are correlated, where λ sets a balance between the neural network's performance and the degree of decorrelation.

B. Data scope training

To increase the significance and decorrelate the neural network output with the reconstructed mass for the background, the loss terms are calculated with different data scopes during training:

$$\begin{aligned} \mathrm{Loss} = &\ \underset{\substack{y = [0,1] \\ \hat{y} \in \text{narrow mass window}}}{\mathrm{Loss}_{\mathrm{classifier}}(\hat{y}, y)} \\ &+ \lambda \cdot \underset{\substack{y = 0 \\ \hat{y},\, \mathrm{mass} \in \text{wide mass window}}}{\mathrm{DisCo}(\mathrm{mass}, \hat{y}).} \end{aligned} \tag{6}$$

The classifier loss, such as BCE loss, is calculated in the scope of signal and background events that fall in the narrow mass window, where the majority of the signal lies. This narrow scope decorrelates the network's output with the mass and can improve the significance. The DisCo term is calculated in the scope of background events that fall in the wide mass window, where the mass window is used in estimating the amount of background in the bump hunt method.

C. Significance-based model selection

When training networks, the best-trained network among the training epochs can be chosen based on an evaluation metric. The loss is often used as the evaluation metric, but the network that has the minimum loss can be different from the network that has the best significance [3,16]. In this work, the expected significance is used as a metric to select the best network among the epochs.

The significance is calculated with the following steps:

(1) Divide the dataset into bins based on the neural network's output. The bins are constructed to have an equal number of signal events.

(2) Calculate the significance of each bin.

(3) Combine the significances of the bins.

The significance [3] of each bin is calculated with

$$\mathrm{Significance} = \sqrt{2\left[(N_S + N_B)\ln\left(1 + \frac{N_S}{N_B}\right) - N_S\right]}, \tag{7}$$

where N_S is the number of signal events and N_B is the number of background events within a mass window containing the 5th to 95th percentile of the signal. Note that the mass window width can change depending on the bin. The combination of significance [17] over the bins is calculated with

$$\text{Total significance} = \sqrt{\sum_i^n (\mathrm{Significance}_i)^2}, \tag{8}$$

where n is the number of bins and i is the index of the bin.

IV. EXAMPLE ANALYSIS

In this work, a search for the process $H \to Z(\ell^+\ell^-)\gamma$ is considered, where $\ell^+\ell^-$ represents an e^+e^- or $\mu^+\mu^-$ pair.

Such a study has been performed by CMS [18] and ATLAS [19] with the Run 2 LHC data using a luminosity of around 150 fb^{-1}, where the BDT technique was used in binning the search region. The expected significance for this standard model process of each analysis is 1.2σ [6,20]. When adding the expected Run 3 LHC data [21], where 250 fb^{-1} of data could be collected, and assuming similar analysis sensitivity, the expected significance is 2σ. To reach a 3σ significance, an additional 600 fb^{-1} of data is required. This will be achieved in the High Luminosity LHC [22] era that is planned to start in 2029, where around 300 $\mathrm{fb}^{-1}/\mathrm{year}$ of data is expected. To reduce the amount of time to obtain evidence (3σ) of this decay, a neural network approach is explored.

The simplified $H \to Z(\ell^+\ell^-)\gamma$ analysis in this work searches for a resonance in the reconstructed mass distribution of the Higgs boson candidates. To increase the significance, the search region of the analysis is binned with specially trained neural networks, and the performance is compared with that of boosted decision trees.

The following sections describe the dataset, input features for the machine-learning techniques, the machine-learning techniques themselves, evaluation metrics, experiment, and results.

A. Dataset

The dataset is generated with the Monte Carlo event generator MadGraph5_AMC@NLO [23,24], particle simulator PYTHIA8 [25], and detector simulator DELPHES3 [26], where jet clustering is performed by the FastJet [27] package. For the event generation of the signal, the Higgs boson $(pp \to H)$ is generated with MadGraph5_AMC@NLO using the Higgs effective field theory (HEFT) model [28] and decayed to $H \to Z(\ell^+\ell^-)\gamma$ using PYTHIA8. The background $pp \to Z(\ell^+\ell^-)\gamma$ is generated with MadGraph5_AMC@NLO at leading order. For detector simulation, the CMS detector settings of DELPHES3 are used. To have samples for training, validation, and testing, a dataset of 4.5×10^7 events is generated for both the signal and the background, for a total of 9.0×10^7 events. The total number of signal events for each of the training, validation, and testing datasets is scaled to have the standard model cross section of 7.52×10^{-3} pb with a luminosity of 138 fb^{-1}, corresponding to the Run 2 luminosity of CMS. The background is scaled to have the standard model cross section of 55.5 pb with the same luminosity as the signal.

The following requirements, which are similar to the CMS analysis [6], are applied to the dataset:

(1) Trigger threshold requirements: $p_T^{\text{leading}\ell} >$ 25 GeV, $p_T^{\text{subleading}\ell} > 15$ GeV.
(2) Background suppression requirements: $p_T^\gamma/m_{\ell\ell\gamma} > 15/110$, minimum $\Delta R(\ell^\pm, \gamma) > 0.4$, $m_{\ell\ell} >$ 50 GeV, $m_{\ell\ell\gamma} + m_{\ell\ell} > 185$ GeV.
(3) Mass window requirement: $100 < m_{\ell\ell\gamma} < 180$ GeV.

After applying these requirements, there are 9×10^6 events for the signal and 5×10^6 events for the background. The large sample helps to reduce overtraining, where the significance metric is sensitive to the sample region that has high classifier scores and a low number of background events.

B. Input features for machine-learning techniques

The following features are used as inputs for the machine-learning techniques that bin the analysis:

(1) η_γ, $\eta_\ell^{\text{leading}\ell}$, $\eta_\ell^{\text{subleading}\ell}$: Pseudorapidity angle of the photon and leptons.
(2) Minimum $\Delta R(\ell^\pm, \gamma)$, maximum $\Delta R(\ell^\pm, \gamma)$: Minimum and maximum ΔR between leptons and the photon.
(3) Flavor of ℓ: Flavor of lepton used to reconstruct the Z boson, either being an electron or a muon.
(4) $p_T^{\ell\ell\gamma}/m_{\ell\ell\gamma}$: p_T of the reconstructed Higgs boson candidate divided by the mass.
(5) $p_{Tt}^{\ell\ell\gamma}$: Projection of the reconstructed Higgs boson p_T to the dilepton thrust axis [29].
(6) $\sigma_m^{\ell\ell\gamma}$: Mass reconstruction error of the $\ell\ell\gamma$ candidate estimated by binning signal events in η and p_T for the photon and leptons, and measuring the signal's mass width for each bin.
(7) $\cos\Theta$, $\cos\theta$, ϕ: Production and decay angles that determine the differential cross section of $H \to Z(\ell^+\ell^-)\gamma$ and $qq \to Z(\ell^+\ell^-)\gamma$ [30,31].
(8) $p_T^\gamma/m_{\ell\ell\gamma}$, $p_T^{\text{leading}\ell}$, $p_T^{\text{subleading}\ell}$: Momenta of the photon and leptons.

Many of the features are correlated with the reconstructed Higgs mass. Therefore, depending on which features are included in the machine-learning technique and which features the machine-learning technique prioritizes, the output of the machine-learning classifier can also be correlated with the mass. Especially when including the momenta of the photon and lepton in certain machine-learning techniques, such as BDTs, to bin the search region, the background mass distribution in certain bins tends to peak close to the Higgs boson mass, which can be seen in Fig. 4. This behavior introduces difficulties in estimating the number of signal events, so that these features are typically excluded for the inputs of the machine-learning technique. However, these features can be used as inputs for neural networks that are trained to be decorrelated with the reconstructed mass.

C. Machine-learning techniques

The following machine-learning techniques are compared:

(1) Boosted decision trees using the TMVA framework [32], which has 850 trees, with a minimum node size of 2.5%.
(2) XGBoost [33], with 100 boosting rounds with a maximum depth of 3. The BDT was optimized by

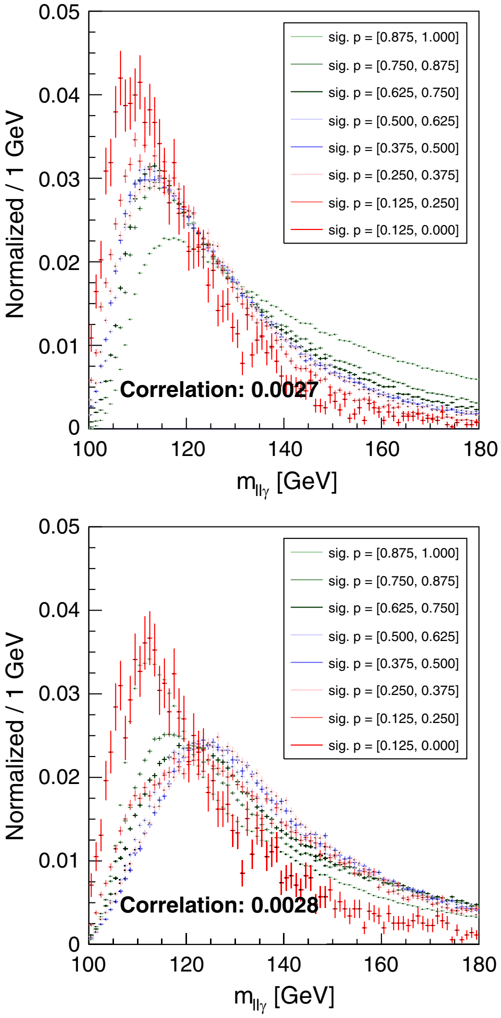

FIG. 4. Top: reconstructed $m_{\ell\ell\gamma}$ background distributions, where each histogram is a bin in the XGBoost output distribution with an equal number of signal events. Lower signal percentile (sig. p) values correspond to higher output values. $p_T^\gamma/m_{\ell\ell\gamma}$, $p_T^{\text{leading}\ell}$, and $p_T^{\text{subleading}\ell}$ are not used in the training. Bottom: reconstructed $m_{\ell\ell\gamma}$ background distributions, when training includes $p_T^\gamma/m_{\ell\ell\gamma}$, $p_T^{\text{leading}\ell}$, and $p_T^{\text{subleading}\ell}$ inputs for XGBoost. For certain bins, the background peaks close to the Higgs boson mass of 125 GeV, which introduces difficulties in estimating the number of signal events. "Correlation" represents the magnitude of difference in the shapes between the machine-learning bins.

modifying the number of boosting rounds, where BDTs with more than 100 rounds have higher overtraining, while other performance metrics are similar.

(3) A feed-forward neural network that has an input layer with N nodes corresponding to the number of input features, a hidden layer with $4N$ nodes with a tanh activation function, and an output layer with one node with a sigmoid activation function. The network is implemented with PyTorch [34].

(4) A deep feed-forward neural network that has an input layer with N nodes corresponding to the number of input features and three hidden layers with $3N$, $9N$, and $3N$ nodes with a tanh activation function. The last hidden layer is connected to a dropout layer with a rate of 0.1. The output layer has a single node with a sigmoid activation function. The network is implemented with PyTorch. The network was optimized by modifying the dropout rate, where the network with no dropout had significant overtraining and poorer performance, while the network with a dropout rate of 0.2 had similar performance to the dropout 0.1 network.

(5) The event classifier transformer with $N_{\text{repr}} = 16$. Each event feature is embedded by a separate feed-forward network that has an input layer with one node, a hidden layer with 4 nodes with a GELU activation function, and an output layer with 16 nodes. The MHA has 4 heads, and the positionwise feed-forward network has an input layer with 16 nodes, a hidden layer with 64 nodes with a GELU activation function, and an output layer with 16 nodes. The dropout layer has a rate of 0.1. The network is implemented with PyTorch. The network was optimized by modifying the dropout rate, where the network with no dropout showed overtraining and poorer performance, while the network with a dropout rate of 0.2 had slightly worse performance than the dropout 0.1 network.

When training the machine-learning techniques, the signal and background samples are weighted to have the same number of events. All input features are normalized for neural networks.

For BDTs and networks that are trained without DisCo loss, 12 input features are used: η_γ, $\eta_\ell^{\text{leading}\ell}$, $\eta_\ell^{\text{subleading}\ell}$, minimum $\Delta R(\ell^\pm,\gamma)$, maximum $\Delta R(\ell^\pm,\gamma)$, flavor of ℓ, $p_T^{\ell\ell\gamma}/m_{\ell\ell\gamma}$, $p_{Tt}^{\ell\ell\gamma}$, $\sigma_m^{\ell\ell\gamma}$, $\cos\Theta$, $\cos\theta$, and ϕ. These input features have minimal correlation with the reconstructed $m_{\ell\ell\gamma}$. The training is performed with events that have a reconstructed Higgs boson mass range between 120 and 130 GeV.

When networks are trained with DisCo loss, three additional features are used: $p_T^\gamma/m_{\ell\ell\gamma}$, $p_T^{\text{leading}\ell}$, and $p_T^{\text{subleading}\ell}$. The data scope training method described in Sec. III B is used. The narrow mass window is from 120 to 130 GeV, and the wide mass window is from 100 to 180 GeV.

Different combinations of loss functions are explored:

(1) BCE loss.
(2) Extreme loss.
(3) BCE + DisCo loss.
(4) Extreme + DisCo loss.

To increase the stability of the calculation, the extreme loss implementation clamps the neural network output $\hat{y}$ to a range from 0.001 to 0.999. The λ DisCo factor is 10 for the BCE + DisCo loss and 50 for the extreme + DisCo loss to have similar background correlations among networks trained with DisCo.

Each neural network is trained for 1200 training epochs with the Adam optimization algorithm [35] with a learning rate of 10^{-3} and a batch size of 8192. The best model between the epochs is selected by finding the model with highest significance on the validation dataset. To reduce the amount of time in training, the model is evaluated on the validation dataset in five training epoch intervals for the first 50 epochs, and then ten training epoch intervals for the remaining training epochs. This method allows the best model search to be sensitive to the early epochs, where the evaluation metric can change dynamically.

D. Machine-learning technique evaluation metrics

The machine-learning techniques are evaluated with the following metrics:

(1) Expected significance.
(2) Area under the curve (AUC) of the receiver operating characteristic (ROC) curve.
(3) Correlation of $m_{\ell\ell\gamma}$ with the machine-learning technique output.
(4) Epoch of the best-trained model.

The significance is calculated using the method described in Sec. III C, where the sample is divided into eight machine-learning technique bins with an equal number of signal events in each bin.

The AUC curve is calculated using the SCIKIT-LEARN [36] package with negative weights set to zero and evaluated in the Higgs boson mass window from 120 to 130 GeV.

The correlation is measured with the following steps:

(1) The search region is binned according to the machine-learning technique output.
(2) Normalized $m_{\ell\ell\gamma}$ histograms are created for each machine-learning technique bin.
(3) The differences between the machine-learning technique bins are measured with the standard deviation of the normalized yield for each mass bin.
(4) The mean of the standard deviations is calculated.

When calculating the correlation, the sample is divided into eight machine-learning technique bins with an equal number of signal events in each bin.

E. Experiment and results

A repeated random subsampling validation procedure [37] is used to evaluate the machine-learning techniques, where three trials are done. The number of trials is limited due to the long training time of the neural networks. The validation procedure follows the steps below:

(1) The events in the dataset are randomly shuffled and divided equally for the training, validation, and test datasets.
(2) The machine-learning technique is trained using the training dataset. For neural networks, the best model between the epochs is selected using the expected significance metric on the validation dataset.
(3) The performance of the machine-learning technique using the evaluation metrics is evaluated on the test dataset.

TABLE I. Average evaluation metrics for machine-learning techniques using the random subsampling evaluation procedure with three trials. “Random” is a random classifier. “FNN” is a feed-forward network. “DFNN” is a deep feed-forward network. “ETN” is an event classifier transformer network. “N.A.” means “not available.” “Ext” is extreme loss. “Signi.” is the expected significance. “AUC” is the area under the curve of the receiver operating characteristic curve. “Bkg. Corr.” is the correlation of $m_{\ell\ell\gamma}$ with the machine-learning technique output for the background calculated with the method described in Sec. IV D. “Best epoch” is the epoch that had the highest significance on the validation dataset. Bold numbers indicate the best values over the machine-learning techniques.

Technique	Loss	Signi.	AUC (%)	Bkg. Corr. (10^{-3})	Best epoch
Random	N.A.	0.50	50.0	0	N.A.
BDT	N.A.	1.17	75.6	3.3	N.A.
XGBoost	N.A.	1.49	75.8	2.7	N.A.
FNN	BCE	1.43	75.2	2.8	22
FNN	Ext	1.45	75.4	2.7	107
DFNN	BCE	**1.50**	**76.0**	2.0	907
DFNN	Ext	1.46	75.7	2.0	487
ETN	BCE	**1.51**	**76.2**	2.3	570
ETN	Ext	1.48	75.7	2.3	307
FNN	BCE + DisCo	1.35	75.0	**0.7**	633
FNN	Ext + DisCo	1.46	74.7	**0.8**	327
DFNN	BCE + DisCo	1.46	75.1	**1.0**	550
DFNN	Ext + DisCo	1.47	75.7	**1.0**	273
ETN	BCE + DisCo	**1.52**	75.6	**1.0**	670
ETN	Ext + DisCo	**1.50**	75.4	**1.0**	623

(4) The steps above are repeated N times, where in this work $N = 3$. The machine-learning technique is reinitialized for each trial.

(5) After all the trials, the evaluation metrics are averaged to assess the performance of the machine-learning techniques.

The average evaluation metrics for the machine-learning techniques are shown in Table I. The experiment results show the following:

(1) The event classifier transformer trained with DisCo loss shows the highest significance with the lowest background correlation between the machine-learning techniques.

(2) The deep feed-forward network and event classifier transformer trained with BCE loss show the highest AUC and significance but have higher background correlations.

(3) The neural networks trained with the DisCo loss show the lowest background correlation.

(4) Networks trained with extreme + DisCo loss are trained more quickly compared to networks trained with BCE + Disco loss, while showing similar performance.

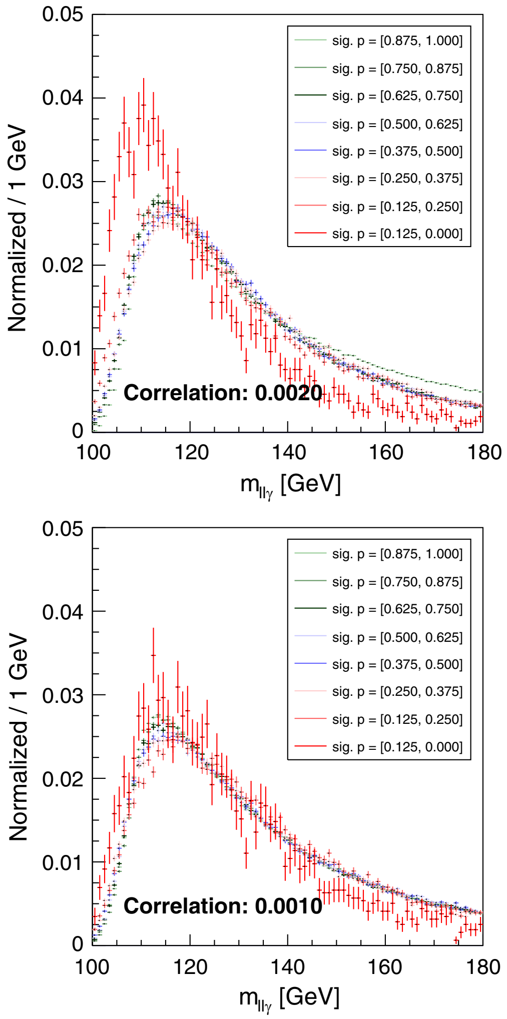

FIG. 5. $m_{\ell\ell\gamma}$ distribution of the background, where each histogram is a bin in the machine-learning technique output distribution. Each bin has an equal number of signal events. Lower signal percentile (sig. p) values correspond to higher network output values. Top: deep feed-forward network trained with BCE loss. Bottom: event classifier transformer network trained with extreme + DisCo loss. Correlation represents the magnitude of difference in the shapes between the machine-learning bins. A lower correlation can be observed with the network trained with DisCo loss.

There is an upper limit on performance for a classifier due to the similarities of the signal and background, where machine-learning classifiers could be optimized in approaching the upper limit in certain phase spaces. Therefore, the goal for the machine-learning classifier is to approach the upper limit in the phase space that is most relevant for the problem. The event classifier transformer tends to be able to approach the upper limit on the significance metric better than the other machine-learning techniques used in this paper. However, with more hypertraining on the other machine-learning techniques or by using different machine-learning techniques, it could be possible to get closer to the upper limit, which is left for future studies.

The machine-learning technique metrics are shown for one of the subsampling trials. The correlation between the $m_{\ell\ell\gamma}$ and network output is shown in Fig. 5 for the deep feed-forward network trained with BCE loss and the event classifier transformer trained with the extreme + DisCo loss. The network output distribution is shown in Fig. 6, where in the figure, overtraining of the network is measured with the χ^2 test [38] implemented in ROOT [39], by comparing the network output distributions on the training sample and validation sample. The residuals in the plot are the normalized differences between the output distributions defined in [38]. For both networks, the bump in the middle of the signal distribution is related with the networks making different output distributions depending on $p_T^{\ell\ell\gamma}/m_{\ell\ell\gamma}$. An effect from the DisCo term can be seen, where the minimum network output value is shifted upwards.

It is noted that when increasing the number of input features, the increase in the number of trainable network weights for the event classifier transformer is small compared to a deep feed-forward network. The smaller increase can make the network easier to train, when more input features are used. For example, when changing the number of input features from 12 to 15, the increase of the number of trainable weights for the event classifier transformer is 265 (a 5% increase), while for the deep feed-forward neural network it is 4,671 (a 55% increase).

During training of the feed-forward network with BCE loss, the BCE loss value and the significance

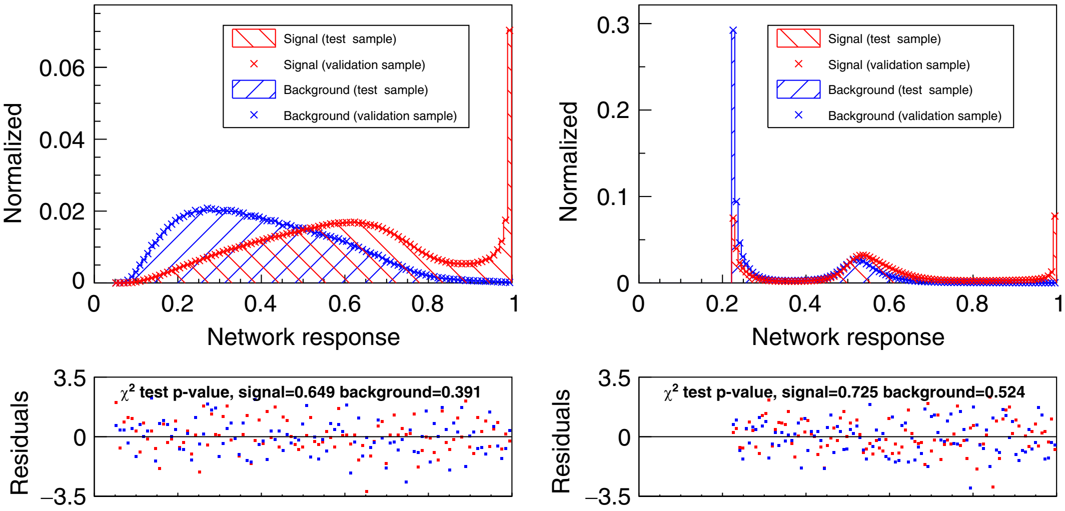

FIG. 6. Left: deep feed-forward network trained with BCE loss. Right: event classifier transformer network trained with the extreme + DisCo loss. Top: network output distribution on the training dataset and validation dataset. Bottom: overtraining of the network evaluated by comparing the network output distribution between the training and validation datasets for the signal and background using a χ^2 test, where residuals are the normalized differences between the output distributions.

can have different trends over the training epochs, which is shown in Fig. 7. This discrepancy demonstrates that when selecting the best model between the epochs, the expected significance should be used instead of the loss.

A study was also performed to compare the results when the data scope training method was not applied. Instead, the wide mass window was used for both the classifier loss and the DisCo loss. The training became difficult where the correlation could be high or the network outputs could converge to one value. The networks that were successfully trained had a few-percent poorer significance when having similar background correlations.

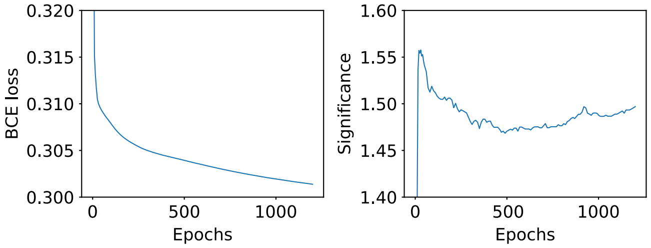

FIG. 7. BCE loss (left) and expected significance (right) over training epochs for a feed-forward neural network trained with BCE loss. The epoch that has the lowest BCE loss does not correspond to the epoch that has the highest significance. The network has a high significance in the beginning of the training, and it becomes lower as the training continues. This behavior is due to the neural network becoming more discriminative for neural network outputs at medium values, but as a tradeoff, it becomes less discriminative for neural network outputs at high values, where the significance dependence is high.

V. RELATED WORK

The Particle Transformer [8] is a transformer-based neural network targeted toward jet-flavor tagging. The structure is similar to the transformer [7], where each particle in a jet is considered as a token. Additionally, variables calculated using the features of a pair of particles are passed through a feed-forward network and added to the attention scores of the MHA. For the final prediction, class tokens are passed as a query to a MHA. The event classifier transformer in this work has a similar structure, where the main difference is that each event feature is embedded to a separate token using separate feed-forward networks, and that the neural network output is used to bin a bump hunt analysis.

There has been work to develop a loss function targeted toward significance, where a loss based on the inverse of the significance [40] has been studied. However when a network is trained with this loss, the neural network outputs are clustered around 0 or 1. This behavior makes the loss unusable for binning a bump hunt analysis. The proposed extreme loss can enhance the significance while not having the network outputs clustered around 0 or 1. It can also reduce the number of epochs that are needed to reach the optimal performance of the network.

DisCo [15] has been used for jet flavor tagging and for the ABCD analysis technique [41], where it has been shown to be effective in decorrelating a targeted feature of the jet with the neural network output and decorrelating two neural network outputs used for the ABCD method. In this work, DisCo is used to decorrelate the neural network output with the reconstructed mass to bin a bump hunting analysis.

VI. SUMMARY AND CONCLUSIONS

A transformer-based neural network is proposed to increase the expected significance of a search for a resonance in a reconstructed mass distribution. The significance is enhanced by binning events using a network that discriminates between signal and background events. To apply the transformer architecture for this task, each event feature is passed through a separate feed-forward network to create tokens for the transformer. This network is called the event classifier transformer.

Special training techniques are proposed to enhance the significance and reduce the correlation between the network's output and reconstructed mass.

(1) Extreme loss is proposed that can enhance the significance and reduce the number of training epochs compared to the commonly used binary cross-entropy loss.

(2) DisCo can be used to reduce the correlation. This allows the network to have input event features that are correlated with the reconstructed mass.

(3) Data scope training is proposed, where loss terms have different data scopes. This method can increase the significance and reduce the correlation.

(4) A significance selection metric is proposed for choosing the best model between the training epochs instead of loss.

In the context of a simplified $H \to Z(\ell^+\ell^-)\gamma$ search, the new event classifier transformer trained with the special techniques shows higher significance and lower mass correlation when compared with boosted decision trees and feed-forward networks. This result demonstrates the potential of the event classifier transformer and the specialized training techniques targeted toward binning a search for a resonance in the reconstructed mass distribution.

ACKNOWLEDGMENTS

I am grateful to Gyuwan Kim in the UCSB Computer Science and Natural Language Processing group for introducing me to the transformer neural network and for revising the paper. I am also grateful to Jeff Richman in the UCSB High Energy Physics group for his guidance and help in editing the paper. The material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics under Award No. DE-SC0011702.

[1] CMS Collaboration, Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC, Phys. Lett. B **716**, 30 (2012).

[2] ATLAS Collaboration, Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC, Phys. Lett. B **716**, 1 (2012).

[3] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, Eur. Phys. J. C **71**, 1554 (2011).

[4] The ATLAS Collaboration, The CMS Collaboration, and The LHC Higgs Combination Group, Procedure for the LHC Higgs boson search combination in Summer 2011, Technical Reports No. CMS-NOTE-2011-005, No. ATL-PHYS-PUB-2011-11, CERN, Geneva, 2011.

[5] CMS Collaboration, Evidence for Higgs boson decay to a pair of muons, J. High Energy Phys. 01 (2021) 148.

[6] CMS Collaboration, Search for Higgs boson decays to a Z boson and a photon in proton-proton collisions at $\sqrt{s} = 13$ TeV, J. High Energy Phys. 05 (2023) 233.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, arXiv:1706.03762v7.

[8] H. Qu, C. Li, and S. Qian, Particle transformer for jet tagging, arXiv:2202.03772v2.

[9] J. L. Ba, J. R. Kiros, and G. E. Hinton, Layer normalization, arXiv:1607.06450v1.

[10] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, arXiv:1512.03385v1.

[11] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jegou, Going deeper with image transformers, arXiv:2103.17239v2.

[12] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929 (2014), https://jmlr.org/papers/v15/srivastava14a.html.

[13] https://github.com/jaebak/EventClassifierTransformer.

[14] I. J. Good, Rational decisions, J. R. Stat. Soc. Ser. B **14**, 107 (1952).

[15] G. Kasieczka and D. Shih, Robust jet classifiers through distance correlation, Phys. Rev. Lett. **125**, 122001 (2020).

[16] M. O. Sahin, D. Krucker, and I.-A. Melzer-Pellmann, Performance and optimization of support vector machines in high-energy physics classification problems, Nucl. Instrum. Methods Phys. Res., Sect. A **838**, 137 (2016).

[17] R. D. Cousins, Annotated bibliography of some papers on combining significances or p-values, arXiv:0705.2209v2.

[18] CMS Collaboration, The CMS experiment at the CERN LHC, J. Instrum. **3**, S08004 (2008).

[19] ATLAS Collaboration, The ATLAS experiment at the CERN Large Hadron Collider, J. Instrum. **3**, S08003 (2008).

[20] ATLAS Collaboration, A search for the Zγ decay mode of the Higgs boson in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector, Phys. Lett. B **809**, 135754 (2020).

[21] S. Fartoukh *et al.*, LHC configuration and operational scenario for run 3, Technical Report No. CERN-ACC-2021-0007, CERN, Geneva, 2021.

[22] G. Apollinari, I. Béjar Alonso, O. Brüning, M. Lamont, and L. Rossi, *High-Luminosity Large Hadron Collider (HL-LHC): Preliminary Design Report*, CERN Yellow Reports: Monographs (CERN, Geneva, 2015).

[23] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations, J. High Energy Phys. 07 (2014) 079.

[24] R. Frederix, S. Frixione, V. Hirschi, D. Pagani, H. S. Shao, and M. Zaro, The automation of next-to-leading order electroweak calculations, J. High Energy Phys. 07 (2018) 185.

[25] C. Bierlich, S. Chakraborty, N. Desai, L. Gellersen, I. Helenius, P. Ilten, L. Lönnblad, S. Mrenna, S. Prestel, C. T. Preuss, T. Sjöstrand, P. Skands, M. Utheim, and R. Verheyen, A comprehensive guide to the physics and usage of PYTHIA 8.3, SciPost Phys. Codebases **2022**, 8 (2022).

[26] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi, DELPHES 3: A modular framework for fast simulation of a generic collider experiment, J. High Energy Phys. 02 (2014) 057.

[27] M. Cacciari, G. P. Salam, and G. Soyez, FastJet user manual, Eur. Phys. J. C **72**, 1896 (2012).

[28] S. Dawson, S. Dittmaier, and M. Spira, Neutral Higgs-boson pair production at hadron colliders: QCD corrections, Phys. Rev. D **58**, 115012 (1998).

[29] M. Vesterinen and T. Wyatt, A novel technique for studying the Z boson transverse momentum distribution at hadron colliders, Nucl. Instrum. Methods Phys. Res., Sect. A **602**, 432 (2009).

[30] J. S. Gainer, K. Kumar, I. Low, and R. Vega-Morales, Improving the sensitivity of Higgs boson searches in the golden channel, J. High Energy Phys. 11 (2011) 027.

[31] J. S. Gainer, W.-Y. Keung, I. Low, and P. Schwaller, Looking for a light Higgs boson in the $Z\gamma \to \ell\bar{\ell}\gamma$ channel, Phys. Rev. D **86**, 033010 (2012).

[32] A. Hoecker *et al.*, TMVA: Toolkit for multivariate data analysis, arXiv:physics/0703039v5.

[33] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, arXiv:1603.02754v3.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., Red Hook, NY, 2019), pp. 8024–8035.

[35] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980v9.

[36] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux, API design for machine learning software: Experiences from the scikit-learn project, arXiv:1309.0238v1.

[37] R. R. Picard and R. D. Cook, Cross-validation of regression models, J. Am. Stat. Assoc. **79**, 575 (1984).

[38] N. D. Gagunashvili, Comparison of weighted and unweighted histograms, arXiv:physics/0605123v1.

[39] R. Brun and F. Rademakers, ROOT: an object oriented data analysis framework, Nucl. Instrum. Methods Phys. Res., Sect. A **389**, 81 (1997).

[40] A. Elwood and D. Krucker, Direct optimisation of the discovery significance when training neural networks to search for new physics in particle colliders, arXiv:1806.00322v1.

[41] G. Kasieczka, B. Nachman, M. D. Schwartz, and D. Shih, Automating the ABCD method with machine learning, Phys. Rev. D **103**, 035021 (2021).