



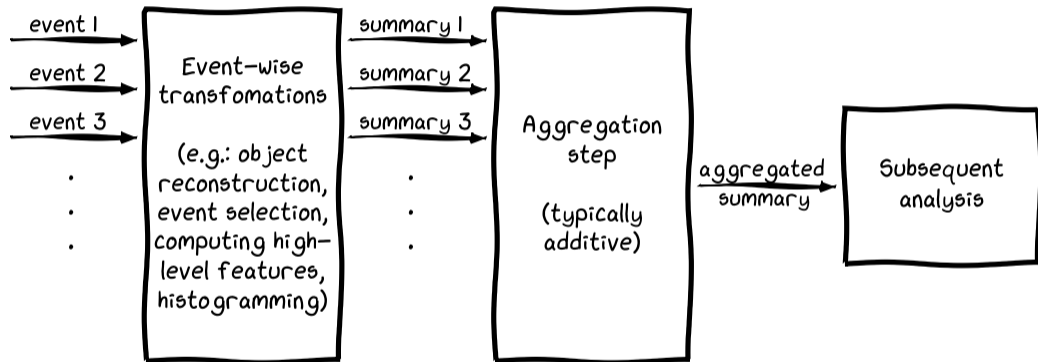
Event2vec: Sensitive vector representations of collider events

<https://github.com/nsmith-/event2vec>

Prasanth Shyamsundar, Fermi National Accelerator Laboratory
(based on work with **Nick Smith**)

SBI Blueprint Workshop, CERN (February 26-27, 2025)

The anatomy of typical HEP analysis



Additive aggregation of event summaries

$$\underbrace{S(D) = \sum_{i=1}^N s(x_i)}_{\text{scalar summary}}$$

$$\underbrace{S_k(D) = \sum_{i=1}^N s_k(x_i)}_{\text{vector summary}}$$

$$\underbrace{S(\theta; D) = \sum_{i=1}^N s(\theta; x_i)}_{\text{function summary}}$$

Examples:

- ▶ **Scalar summary:** cut and count analysis. $s(x_i) \in \{0, 1\}$.
- ▶ **Vector summary:** histogram analysis. $s_b(x_i) = \begin{cases} 1, & \text{if } x_i \in \text{bin-}b, \\ 0, & \text{otherwise.} \end{cases}$
- ▶ **Function summary:** $s(\theta; x_i)$ is the log-likelihood function.

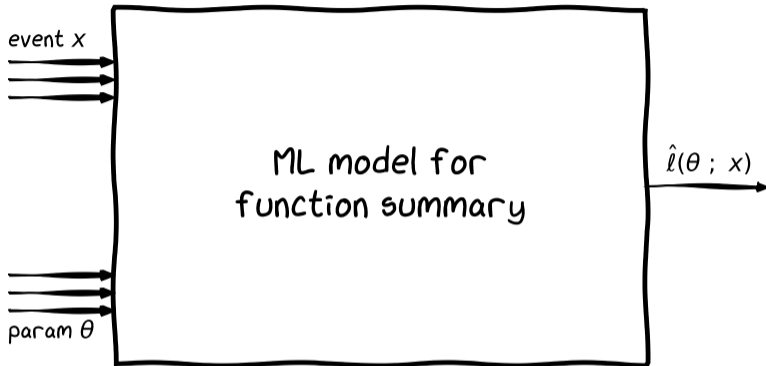
Big-ish picture

- ▶ **Goal:** Construct sensitive vector summaries for EFT analyses, as an alternative to the standard SBI approach.
- ▶ **Motivation:** Statistical analysis in the standard SBI approach is difficult. It is unclear how to account for mc-stat uncertainties both consistently and relatively computationally inexpensively.
- ▶ **Hope:** This will lead to a technique that will be
 - on par with standard SBI in terms of sensitivity,
 - significantly simpler to perform statistical analyses for.

Two approaches

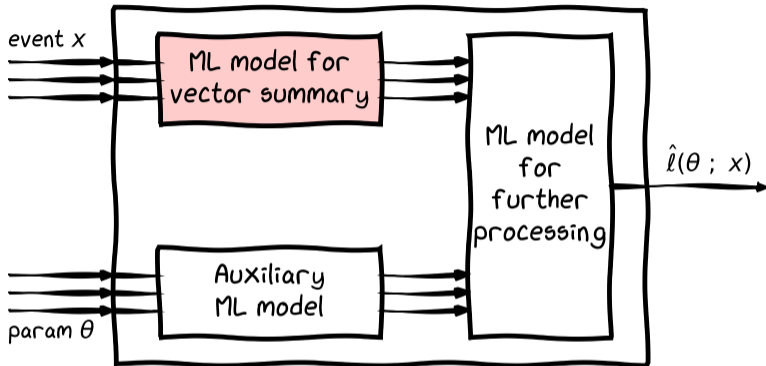
1. Learn a function summary/likelihood ratio func
→ subsequently discretize
2. Or alternatively...

Function summary to a vector summary



- ▶ Train $\hat{\ell}(\theta; x)$ to predict the likelihood or the log-likelihood function (using MadMiner-esque methods).
- ▶ Extract the trained vector summary model, and use it as a sensitive observable.
- ▶ How to analyze the vector summary? (Bin and histogram? Not what we want.)

Function summary to a vector summary



- ▶ Train $\hat{\ell}(\theta; x)$ to predict the likelihood or the log-likelihood function (using MadMiner-esque methods).
- ▶ Extract the trained vector summary model, and use it as a sensitive observable.
- ▶ How to analyze the vector summary? (Bin and histogram? Not what we want.)

Non-probabilistic regular vector summary

- ▶ Let $\ell(\theta ; x)$ be the event-wise log-likelihood function $\ln P(x ; \theta)$.
- ▶ Model an estimator for $\ell(\theta ; x)$ as follows:

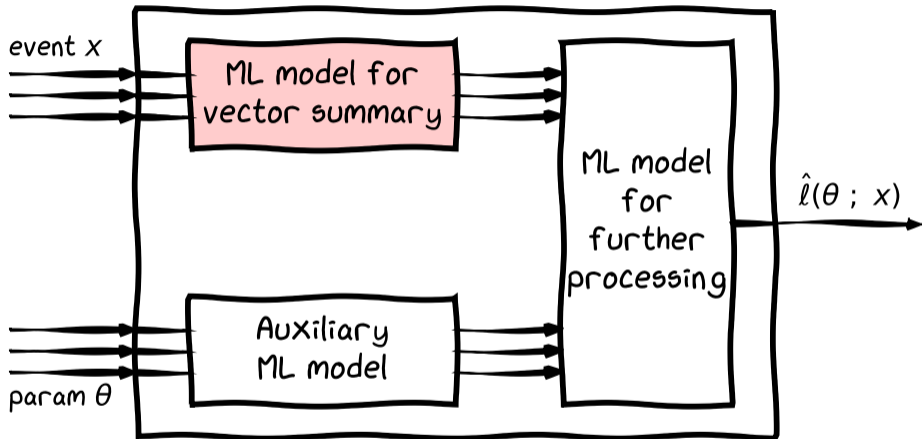
$$\ell(\theta ; x) \stackrel{\text{est}}{\approx} \hat{\ell}_{\varphi}(\theta ; x) \equiv \vec{s}_{\varphi}(x) \cdot \vec{\beta}_{\varphi}(\theta).$$

- ▶ After training φ , the dataset-wise log-likelihood function can be approximated as

$$\ln P(D ; \theta) \equiv \sum_{i=1}^N \ell(\theta ; x_i) \stackrel{\text{est}}{\approx} \underbrace{\left[\sum_{i=1}^N \vec{s}_{\varphi}(x_i) \right]}_{\text{additive aggregation!}} \cdot \vec{\beta}_{\varphi}(\theta).$$

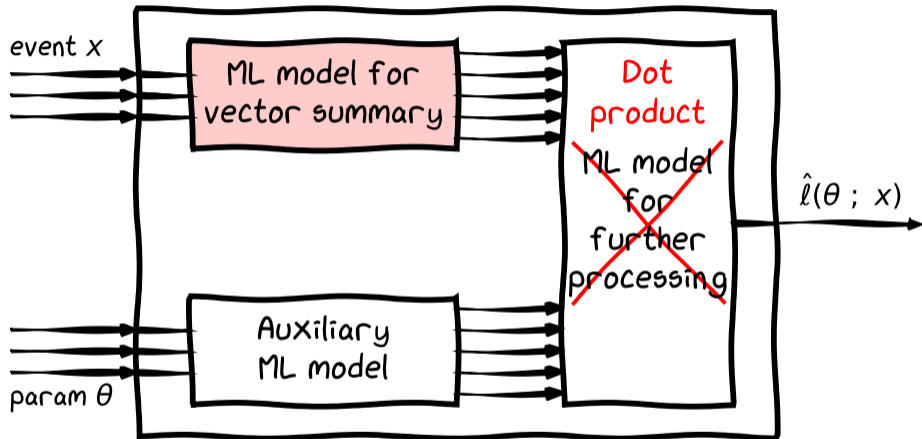
- ▶ So, $\vec{s}_{\varphi}(x)$ is an event-representation amenable to additive aggregation.

Non-probabilistic regular vector summary



As before, first train $\hat{\ell}(\theta; x)$, then extract and use summary model s_φ .

Non-probabilistic regular vector summary



As before, first train $\hat{\ell}(\theta; x)$, then extract and use summary model s_φ .

Probabilistic one-hot constant magnitude summary

- ▶ What if we just want a histogram, not a *generalized* histogram? But we don't want to manually bin some observable.
- ▶ Say we want to specify the number of bins and have an ML-model tell us which bin to put an event in, i.e., we want the machine to automatically dice up the phase-space.

Some options for ML-modeling histograms:

- ▶ Parameterize the high-dimensional phase-space cuts (not used).
- ▶ **Make the map from an event to a bin probabilistic.**

Details:

- ▶ Let $p_{\varphi}^{(b)}(x)$ be the probability of assigned event x to bin b .
- ▶ Let $\hat{\ell}_{\varphi}^{(b)}(\theta)$ be a prediction for the log-likelihood of θ , based only on the index b .
- ▶ Train both functions simultaneously, then use p_{φ}^b .

Cost functions

- ▶ Let $\mathcal{L}_{\text{binary-class}}$ be a binary classification loss function.
- ▶ Cost function for standard SBI training:

$$\mathcal{C}(\varphi) \equiv \mathbb{E}_{(x,w,y,\theta_0,\theta_1) \sim P_{\text{train}}} \left[w \mathcal{L}_{\text{binary-class}} \left(\hat{\ell}_{\varphi}(\theta_1; x) - \hat{\ell}_{\varphi}(\theta_0; x), y \right) \right].$$

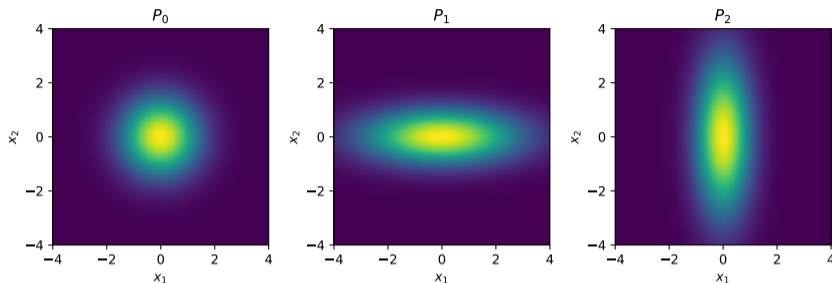
- ▶ Cost function for training a regular vector summary:

$$\mathcal{C}(\varphi) \equiv \mathbb{E}_{(x,y,\theta_0,\theta_1) \sim P_{\text{train}}} \left[w \mathcal{L}_{\text{binary-class}} \left(\vec{s}_{\varphi}(x) \cdot \left(\vec{\beta}_{\varphi}(\theta_1) - \vec{\beta}_{\varphi}(\theta_0) \right), y \right) \right].$$

- ▶ Cost function for training a one-hot vector summary:

$$\mathcal{C}(\varphi) \equiv \mathbb{E}_{(x,y,\theta_0,\theta_1) \sim P_{\text{train}}} \left[\sum_{b=1}^B p_{\varphi}^{(b)}(x) w \mathcal{L}_{\text{binary-class}} \left(\hat{\ell}_{\varphi}^{(b)}(\theta_1) - \hat{\ell}_{\varphi}^{(b)}(\theta_0), y \right) \right].$$

Toy example



- ▶ Mixture of three 2-d Gaussians:

$$P(x_1, x_2; \theta_1, \theta_2) = (1 - \theta_1 - \theta_2) P_0(x_1, x_2) + \theta_1 P_1(x_1, x_2) + \theta_2 P_2(x_1, x_2).$$

- ▶ Excess in left/right regions \leftrightarrow large θ_1
Excess in top/bottom regions \leftrightarrow large θ_2
We'd like our summaries to capture this

event2vec package (<https://github.com/nsmith-/event2vec>)

event2vec

Contributor quick start

1. Clone repo and enter project dir:

```
git clone https://github.com/nsmith-/event2vec.git
cd event2vec
```

2. Create virtual environment (with dev dependencies) and activate it.

- o Using `uv` (preferred; uses `uv.lock`):

```
uv sync
source .venv/bin/activate
```

- o Using pip (ignores lock file):

```
python -m venv .venv --prompt event2vec
source .venv/bin/activate
pip install -e . --group dev
```

Note: If using `uv`, the remaining commands can be run without activating the venv, by prepending each command with `uv run .`

3. Set up the pre-commit git hooks:

```
pre-commit install
```

4. Try the `e2vrun` script:

```
e2vrun --help
```

Gaussian mixture example experiment:

```
e2vrun -o runs/gauss GaussianMixture
```

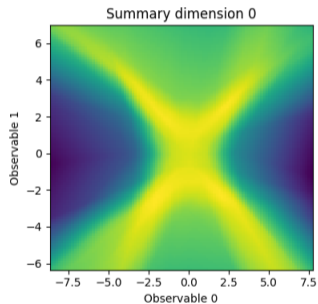
This will create an experiment directory `runs/gauss/` with training logs, model checkpoints, and analysis plots. It runs in a few minutes on one CPU.

To run with binwise loss, use

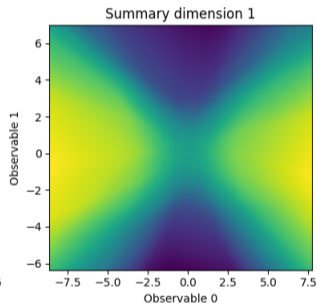
```
e2vrun -o runs/gauss_bin GaussianMixture --loss bce --binwise --summary-dim 3
```

Some results: 2-d regular vector summary

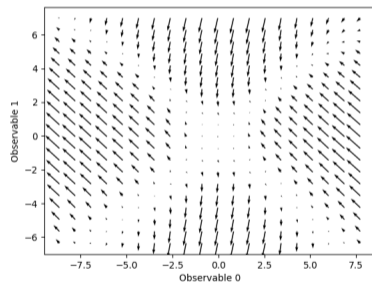
```
e2vrun -o runs/gauss GaussianMixture --epoch 1
```



$\text{summary}_0(x_1, x_2)$



$\text{summary}_1(x_1, x_2)$

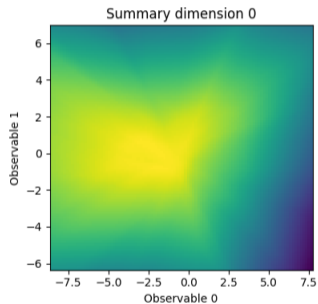


quiver-plot of $\text{summary}(x_1, x_2)$

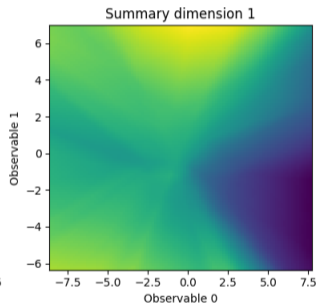
Some results: 2-d regular vector summary

(Before training)

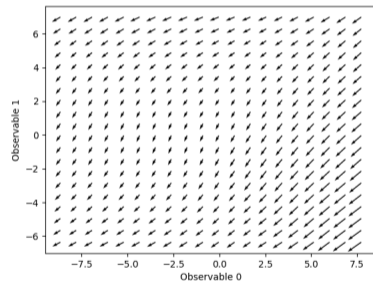
```
e2vrun -o runs/gauss GaussianMixture --epoch 0
```



$\text{summary}_0(x_1, x_2)$



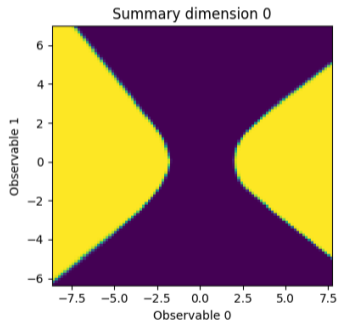
$\text{summary}_1(x_1, x_2)$



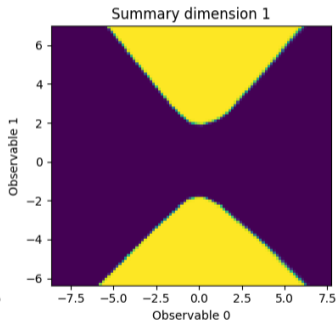
quiver-plot of $\text{summary}(x_1, x_2)$

Some results: One-hot summary

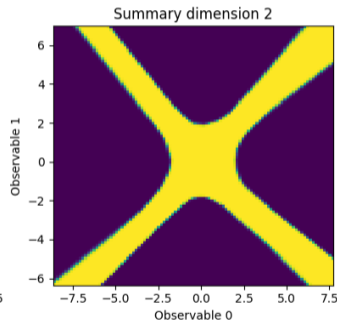
```
e2vrun -o runs/gauss GaussianMixture --summary-dim 3 --binwise --epoch 1
```



$$p^{(0)}(x_1, x_2)$$



$$p^{(1)}(x_1, x_2)$$

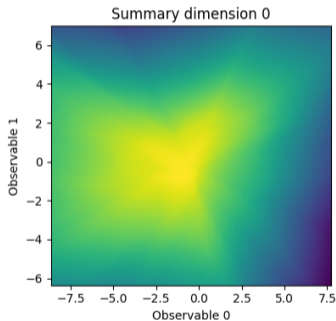


$$p^{(2)}(x_1, x_2)$$

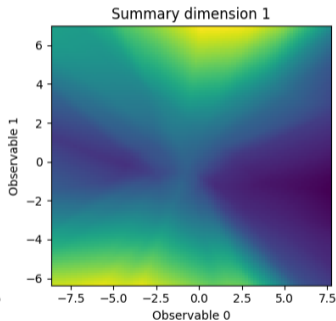
Some results: One-hot summary

(Before training)

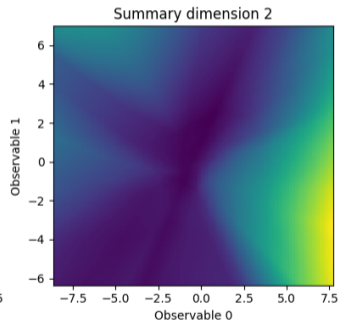
```
e2vrun -o runs/gauss GaussianMixture --summary-dim 3 --binwise --epoch 0
```



$$p^{(0)}(x_1, x_2)$$



$$p^{(1)}(x_1, x_2)$$



$$p^{(2)}(x_1, x_2)$$

Summary

- ▶ Event2vec is a technique (and a tool) for constructing sensitive vector summaries of events, amenable to analysis by additive aggregation.
- ▶ Can be a simpler but effective alternative to the standard SBI approach. (binned doesn't necessarily mean worse)
- ▶ As the number of summary dimensions increases, given enough training data, one can approximate the true likelihood better and better.
- ▶ There are more special sauces in event2vec, in particular for EFT analyses (e.g., PSD matrix regression to exploit underlying structures).
- ▶ We are fleshing out the tool and working on physics examples with a few groups. More to come...

Summary

- ▶ Event2vec is a technique (and a tool) for constructing sensitive vector summaries of events, amenable to analysis by additive aggregation.
- ▶ Can be a simpler but effective alternative to the standard SBI approach. (binned doesn't necessarily mean worse)
- ▶ As the number of summary dimensions increases, given enough training data, one can approximate the true likelihood better and better.
- ▶ There are more special sauces in event2vec, in particular for EFT analyses (e.g., PSD matrix regression to exploit underlying structures).
- ▶ We are fleshing out the tool and working on physics examples with a few groups. More to come...

Thank you! Questions?

Acknowledgments



These slides were prepared using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, Office of High Energy Physics HEP User Facility. Fermilab is managed by Fermi Forward Discovery Group, LLC, acting under Contract No. 89243024CSC000002.

PS is supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics QuantISED program under the following grants:

- ▶ “HEP Machine Learning and Optimization Go Quantum”, Award Number 0000240323
- ▶ “DOE QuantISED Consortium QCCFP-QMLQCF”, Award Number DE-SC0019219