

Moment Pooling:

Gaining Performance and Interpretability Through Physics Inspired Product Structures

Rikab Gambhir

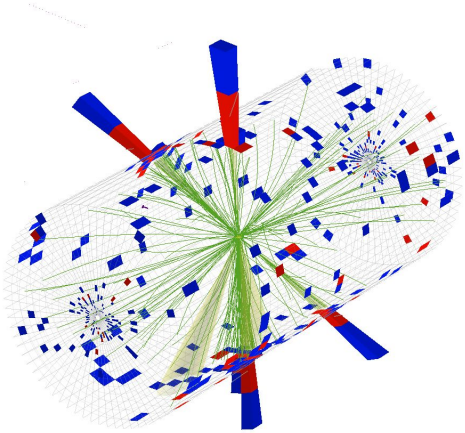
With Athis Osathapan and Jesse Thaler

Email me questions at rikab@mit.edu!

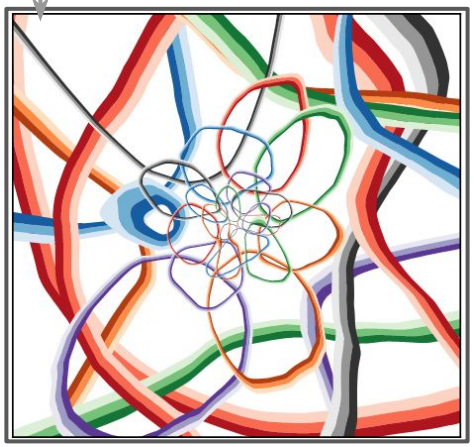
Based on [**RG**, Osathapan, Thaler, 23XX.XXXX (WIP)]

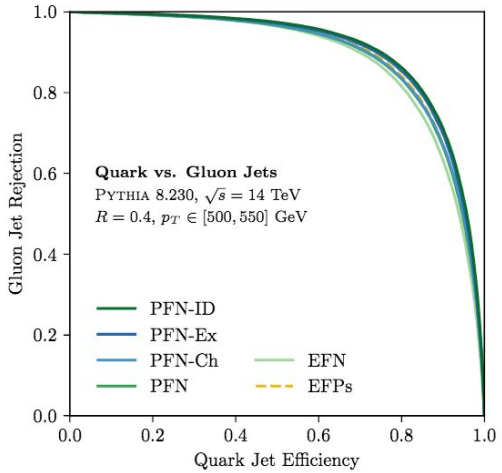
[Komiske, Metodiev, Thaler, 1810.05165]

Typical Machine Learning Setup



Contours of the L different latent representations, ϕ





Collider Data ~1000 Dimensional → **Latent Space** ~10-100 Dimensional → **Observables** ~1-10 Dimensional

Pictured: An Energy Flow Network (**EFN**):

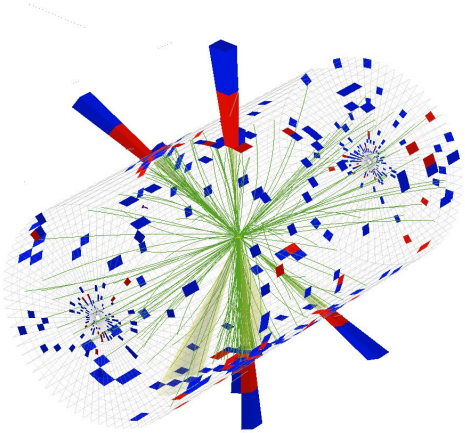
Particle weight: Energy Fraction

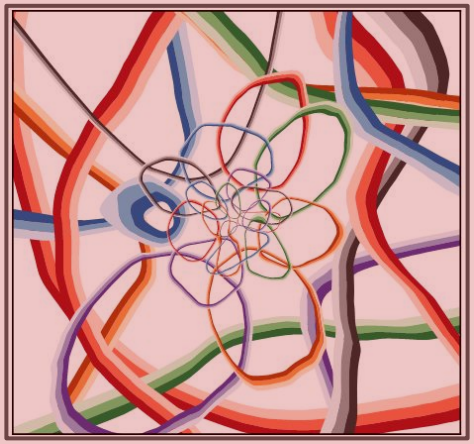
$$\mathcal{O}(\{p_1, \ldots, p_M\}) = F\left(\sum_{i=1}^{M} z_i \Phi(\hat{p}_i)\right)$$

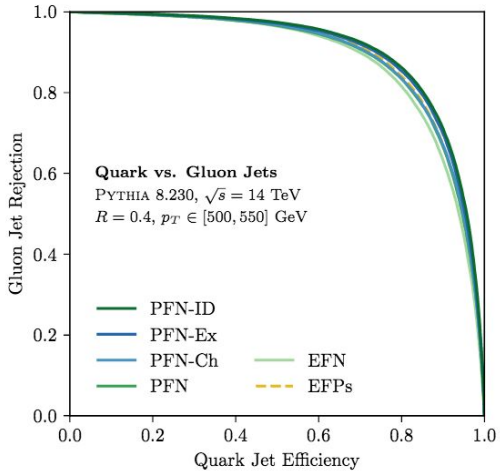
L-dimensional latent representation, per particle

[Komiske, Metodiev, Thaler, 1810.05165]

Typical Machine Learning Setup







Collider Data → **Latent Space** → **Observables**

~1000 Dimensional | ~10-100 Dimensional | ~1-10 Dimensional

Pictured: An Energy Flow Network (**EFN**):

$$\mathcal{O}(\{p_1, \ldots, p_M\}) = F\left(\sum_{i=1}^{M} z_i \Phi(\hat{p}_i)\right)$$

(How) can we understand and constrain this?

(How) can we be more efficient?

The Moment-EFN

$$\mathcal{O}(\mathcal{P}) = F\left(\langle \phi^a \rangle_{\mathcal{P}}\right)$$

EFNs* can be thought of as taking the (weighted) **mean** of a latent particle representation ϕ – Let's generalize to *any* **moment**!

Generalize!

$$\mathcal{O}_k(\mathcal{P}) = F_k\left(\langle \phi^a \rangle_{\mathcal{P}}, \langle \phi^{a_1}\phi^{a_2} \rangle_{\mathcal{P}}, \ldots, \langle \phi^{a_1}\ldots\phi^{a_k} \rangle_{\mathcal{P}}\right)$$

Claim: This "Moment-EFN" gives more efficient representations!

*Most of what I say here today also applies to Particle Flow networks or any other Deep-Sets inspired architecture!

The Moment-EFN (Details)

Set of momenta

a = 1 … *L*, the *Latent Dimension* index

$$\mathcal{O}(\mathcal{P}) = F\left(\langle \phi^a \rangle_{\mathcal{P}}\right)$$

The **Deep Sets Theorem** guarantees that any function on sets $\mathscr{P}$ can be written this way, for sufficiently complex *F* and ϕ

$$\langle \phi \rangle_{\mathcal{P}} \equiv \sum_i z_i \phi(\hat{p}_i)$$

Generalize!

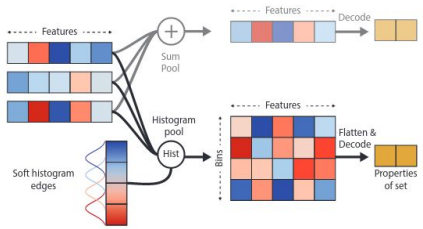
k = Highest order moment considered

$$\mathcal{O}_k(\mathcal{P}) = F_k\left(\underbrace{\langle \phi^a \rangle_{\mathcal{P}}}_{1^{\text{st}}\text{ Moment}}, \underbrace{\langle \phi^{a_1}\phi^{a_2} \rangle_{\mathcal{P}}}_{2^{\text{nd}}\text{ Moment}}, \ldots, \underbrace{\langle \phi^{a_1}\ldots\phi^{a_k} \rangle_{\mathcal{P}}}_{k^{\text{th}}\text{ Moment}}\right)$$

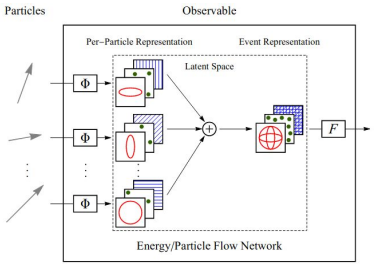
$$L_{\text{eff}} = \frac{k+1}{L}\binom{k+L}{k+1}$$

Claim: This “Moment-EFN” gives more efficient representations!

NSF AI Institute for Artificial Intelligence & Fundamental Interactions



Extending Deep Sets to Distributions
[Histogram Pooling, M. Cranmer et. al.]



IRC-safe Neural Networks
[Energy Flow Networks, 1810.05165]

Moment-EFNs

k = Highest order moment considered

$$\mathcal{O}_k(\mathcal{P}) = F_k\left(\underbrace{\langle\phi^a\rangle_{\mathcal{P}}}_{1^{\text{st}}\text{ Moment}}, \underbrace{\langle\phi^{a_1}\phi^{a_2}\rangle_{\mathcal{P}}}_{2^{\text{nd}}\text{ Moment}}, \ldots, \underbrace{\langle\phi^{a_1}\ldots\phi^{a_k}\rangle_{\mathcal{P}}}_{k^{\text{th}}\text{ Moment}}\right)$$

$$L_{\text{eff}} = \frac{k+1}{L}\binom{k+L}{k+1}$$

Using summary statistics of energy distributions within events to improve performance per parameter

e.g. Jet Angularities

In the moment language, even integer* β jet angularities $\Leftrightarrow$ $k = \beta^{\text{th}}$ moments!

$$\lambda^{(\beta)}(\mathcal{P}) = \sum_i z_i \left(\eta_i^\beta + \phi_i^\beta\right)$$
$$= \langle \eta^\beta \rangle_{\mathcal{P}} + \langle \phi^\beta \rangle_{\mathcal{P}}$$

For the normal (k = 1) EFN, this would require learning nonlinear functions!

Test: Train three networks to regress $\lambda^{(2)}$ from 100k QCD jet samples, with a latent dimension L:

- Linear Network: ϕ, F are 1 layer, linear functions, L = 2
- Small Network: ϕ, F are 2 layers, each with 4 nodes and *LeakyReLU*, L = 2
- "Large" Network: ϕ, F are 3 layers, each with 32 nodes and *LeakyReLU*, L = 8

Expect k = 2 to outperform k = 1 for smaller networks!

*Ask later about non-even or non-integer β angularities

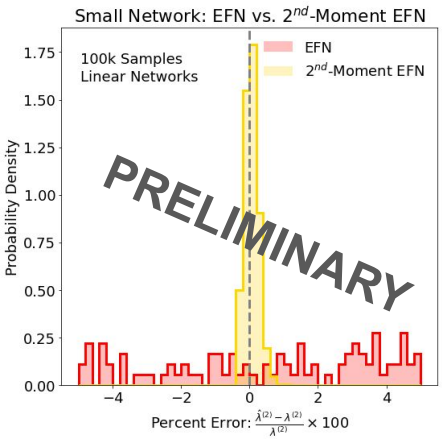
e.g. Jet Angularities

In the moment language, even integer* β jet angularities $\Leftrightarrow$ $k = \beta^{\text{th}}$ moments!

$$\lambda^{(\beta)}(\mathcal{P}) = \sum_i z_i \left(\eta_i^\beta + \phi_i^\beta\right)$$
$$= \left\langle \eta^\beta \right\rangle_{\mathcal{P}} + \left\langle \phi^\beta \right\rangle_{\mathcal{P}}$$

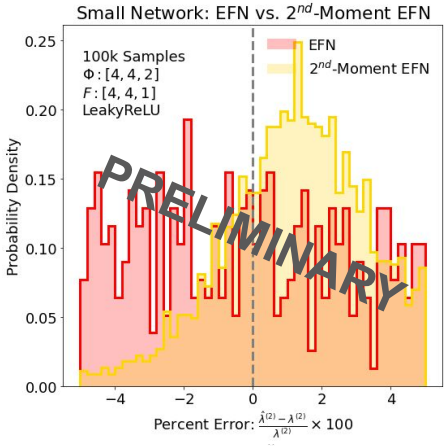
For the normal (k = 1) EFN, this would require learning nonlinear functions!

Linear Networks



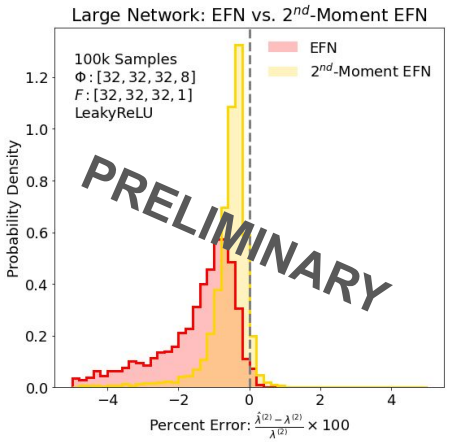
~10 Parameters

Small Networks



~100 Parameters

Large Networks



~6000 Parameters

Training times are identical for k =*1* and *2*!

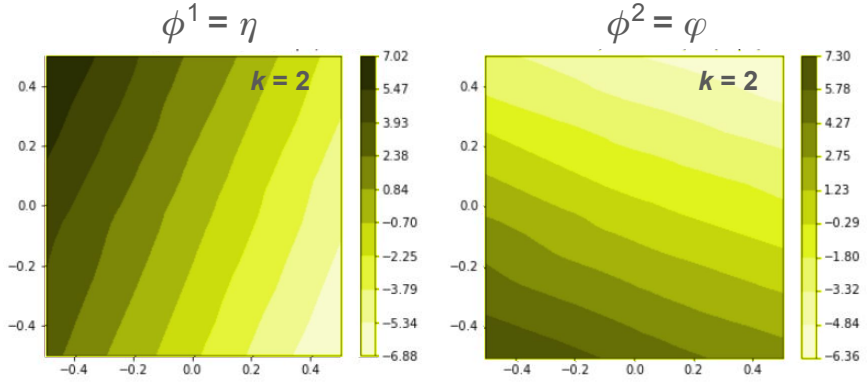
e.g. Jet Angularit

In the moment language, even i

$$\lambda^{(\beta)}(\mathcal{P}) = \sum_i z_i \left(\eta_i^\beta + \phi_i^\beta\right)$$

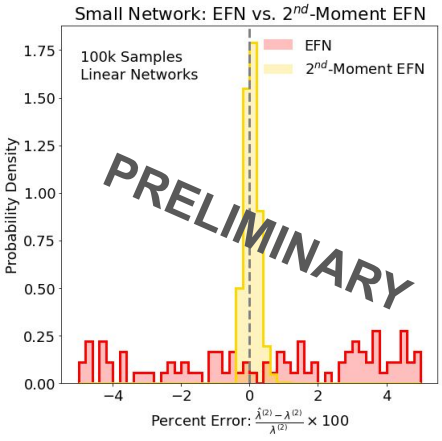
$$= \langle \eta^\beta \rangle_{\mathcal{P}} + \langle \phi^\beta \rangle_{\mathcal{P}}$$

Learns the simplest latent representations!



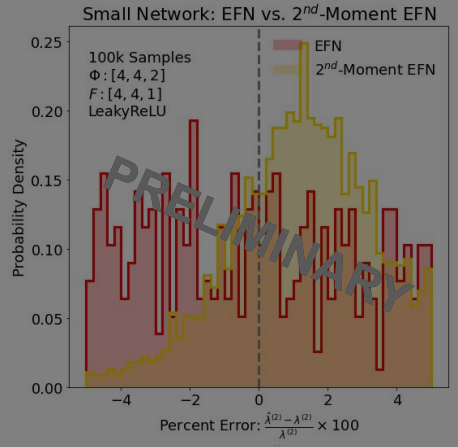
Encourage learning a diagonal basis in (η,φ) using L1 Regularization on *F*

Linear Networks



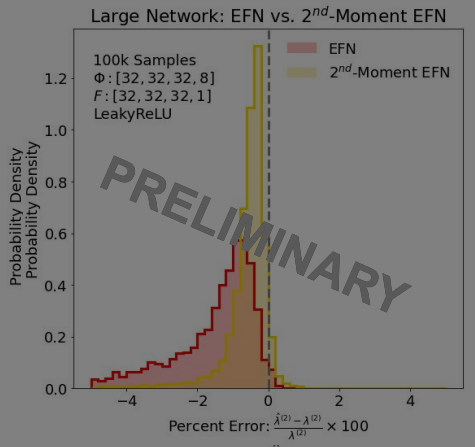
~10 Parameters

Small Networks



~100 Parameters

Large Networks



~6000 Parameters

Training times are identical for *k* =*1* and *2*!

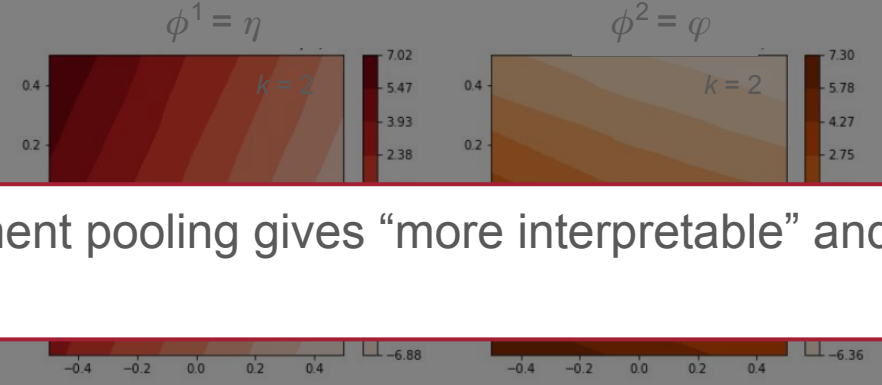
e.g. Jet Angularit

In the moment language, even i

λ

$= \langle n^\beta \rangle + \langle \phi^\beta \rangle_{\mathcal{P}}$

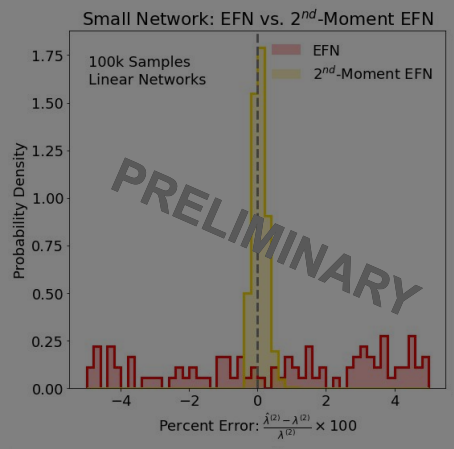
Learns the simplest latent representations!



Encourage learning a diagonal basis in (η,φ) using L1 Regularization on F

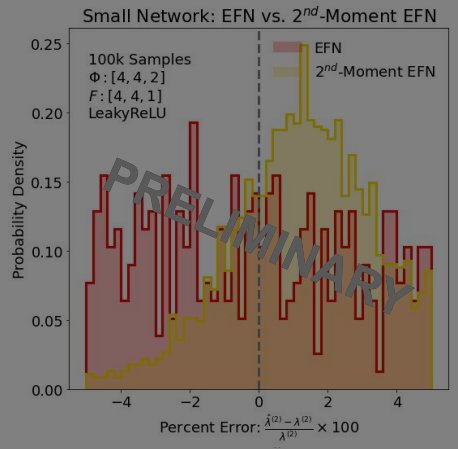
… At least *one* case where moment pooling gives "more interpretable" and "more accurate" networks! ✓

Linear Networks



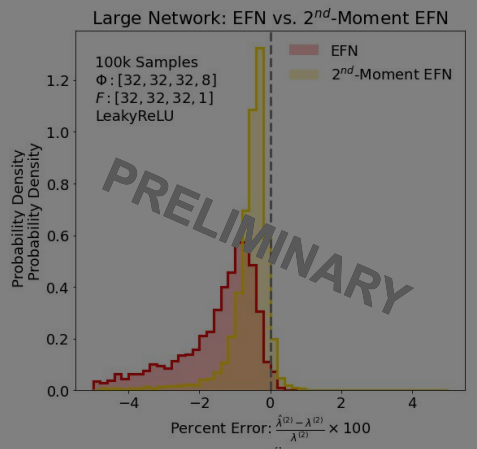
~10 Parameters

Small Networks



~100 Parameters

Large Networks



~6000 Parameters

Training times are identical for *k =1* and *2*!

Linearization

In principle, with a large enough *k*, we can approximate any* observable with a **linear** *F* – like a Taylor expansion on distributions!

$$\begin{aligned}\mathcal{O}(\mathcal{P}) \approx & F_a \langle \phi^a \rangle_{\mathcal{P}} \\ & + F_{a_1 a_2} \langle \phi^{a_1} \phi^{a_2} \rangle_{\mathcal{P}} \\ & + F_{a_1 a_2 a_3} \langle \phi^{a_1} \phi^{a_2} \phi^{a_3} \rangle_{\mathcal{P}} \\ & + \ldots\end{aligned}$$

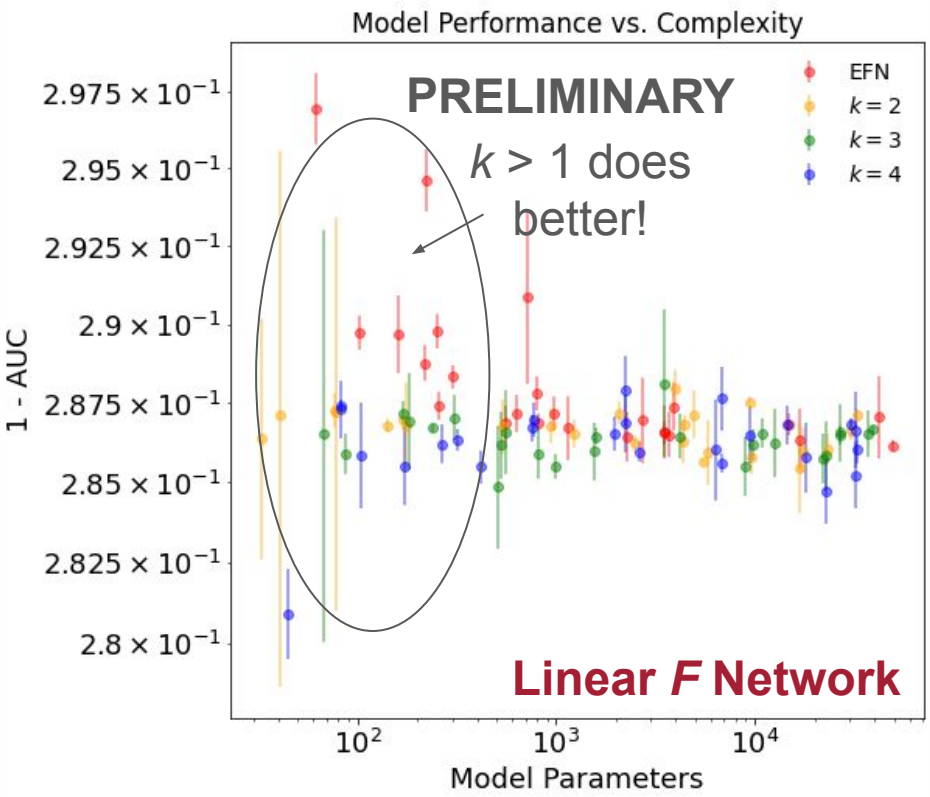
Intuitively: The product structure of moments capture all the nonlinearities that *F* would have captured

Can we understand the behavior of complex observables, like a Q/G discriminant**, with just a few powers of *k*?

*For well-behaved functions

**For classification tasks, we linearize the log likelihood and apply a sigmoid or softmax at the end

e.g. Quark/Gluon Discrimination



Dataset of 500k Quark/Gluon Jets

Series of networks where ϕ has 3 layers and F is linear, but the width of ϕ and the latent dimension is sampled over – **ensemble of networks**

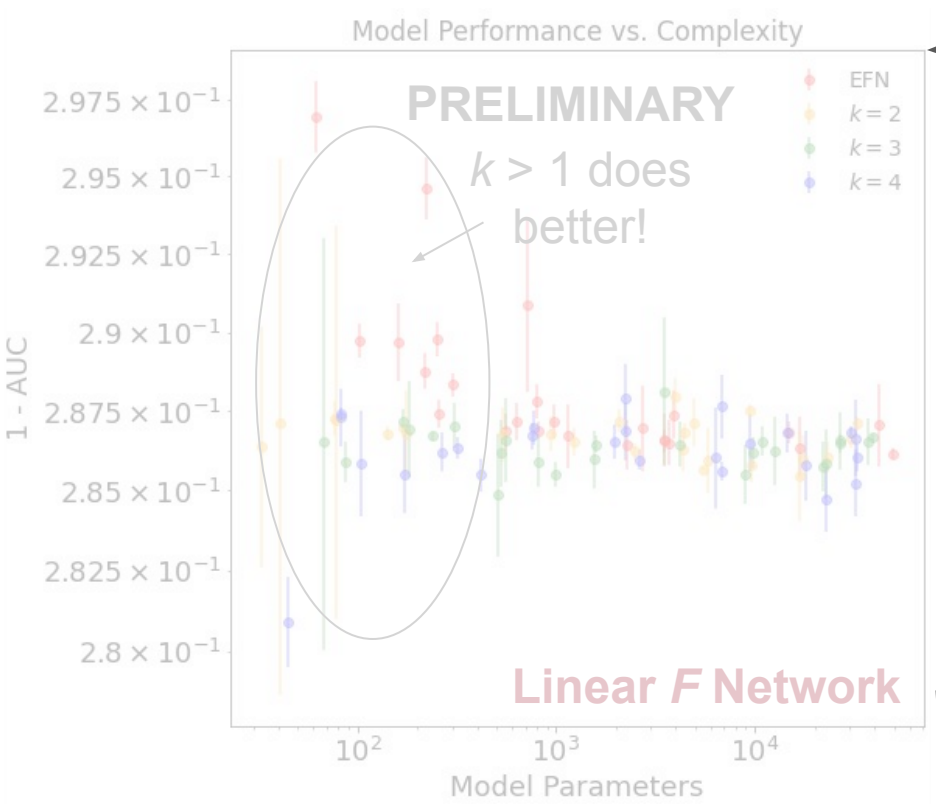
Consider k = 1, 2, 3, and 4

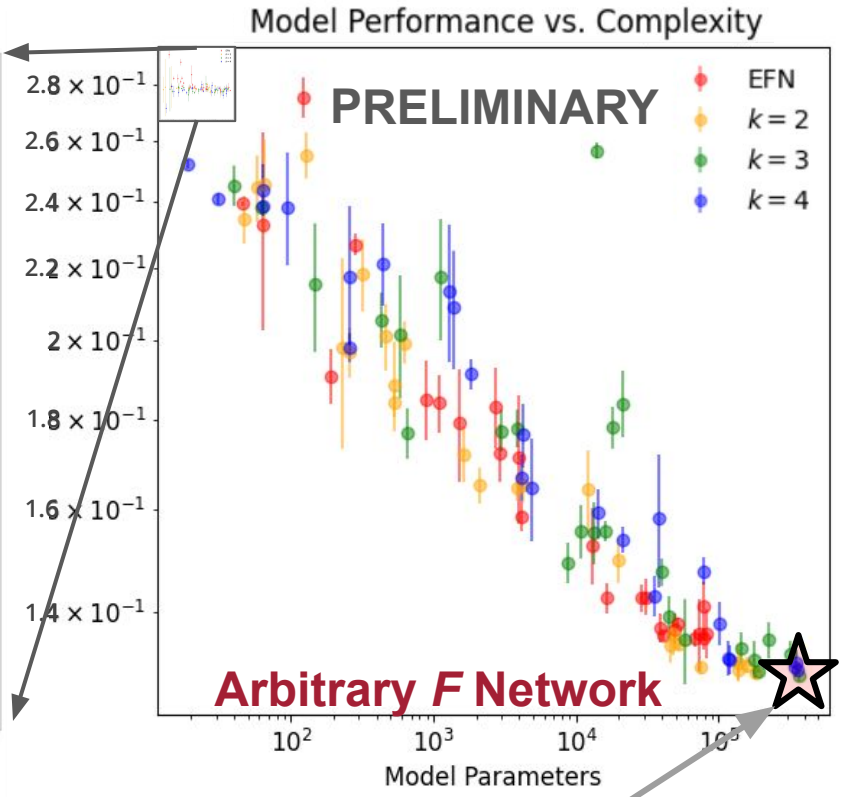
As with the previous example: Moments help achieve the same performance for less parameters!

See Backup slides for training details and dataset details

e.g. Quark/Gluon Discrimination

Compare to an arbitrary *F* network …





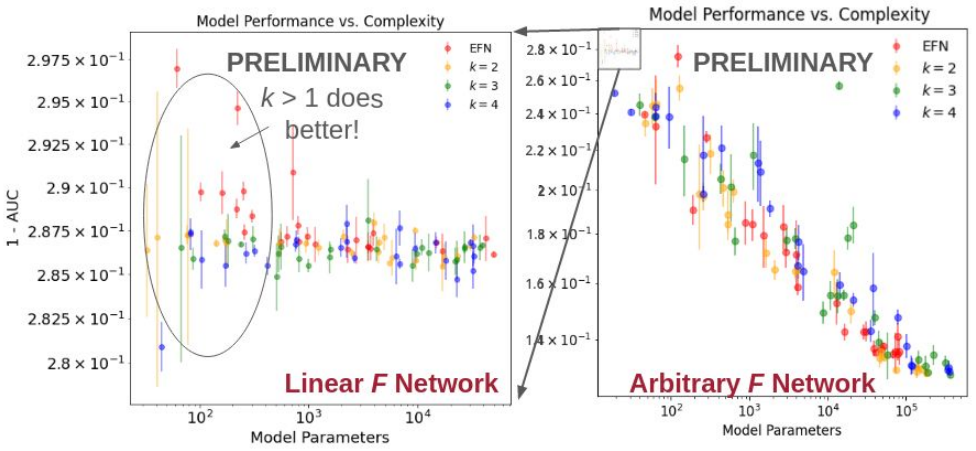
Saturates at AUC = 0.88, consistent with [1810.05165](1810.05165)!

See backup slides for training details and dataset details.
If we have time – see backup for performance versus latent dimension! Same performance for *lower* latent dimensions!

[Komiske, Metodiev, Thaler, 1712.07124;
Cal, Thaler, Waalewijn, 2205.06818]

Interpretation

The linear approximation is not good enough here, unlike jet angularities …



Four moments are not enough to build a linear approximation

⇔

Q/G discrimination cannot be easily captured by low-order moments! No simple closed-form approximation

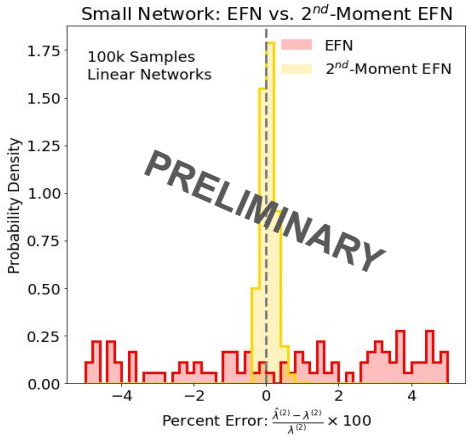
Comparable with the story of Energy Flow Polynomials (EFPs): Degree 4 EFPS achieve an AUC ~ 0.75, but need d > 7 to achieve an AUC above 0.8

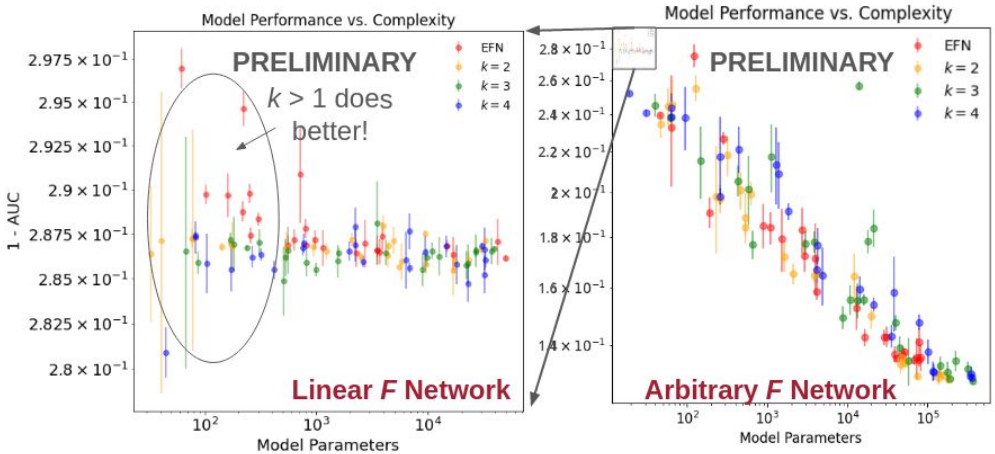
Conclusion

$$\mathcal{O}_k(\mathcal{P}) = F_k\left(\langle\phi^a\rangle_{\mathcal{P}}, \langle\phi^{a_1}\phi^{a_2}\rangle_{\mathcal{P}}, ..., \langle\phi^{a_1}...\phi^{a_k}\rangle_{\mathcal{P}}\right)$$

The Moment-Pooling structure improves open Deep Sets-type networks – better performance for smaller networks!

Can characterize how "simple" or "interpretable" an observable is by how many moments it takes to capture it in a linear basis – angularities are interpretable, Q/G discriminants may not be!





Appendices

Angularities Study (Details)

Dataset:

- 14 TeV Z+jet[g, uds] events generated in Pythia 8.226
- Jets clustered using AK4 (Fastjet 3.3.0)
- Keep p_T between 500 GeV and 550 GeV, $|y| < 1.7$
- 100k Train, 2.5k Val, 2.5k Test
- Angularities normalized to unit mean and standard deviation
- Particle p_T normalized to one.

Training:

- Batch Size: 512
- Epochs: 100
- Optimizer: Adam with learning rate 0.001

Q/G Study (Details)

Same dataset as angularity study, but with 500k training samples

For each of $k = 1 \ldots 4$:

1. Choose random integers F_{size} and ϕ_{size} from 1 … 128, and L from 1 … L_{max}, where L_{max} = 128 for $k = 1$, 64 for $k = 2$, 32 for $k = 3$, and 16 for $k = 4$.
   a. Choose such that the number of network parameters is uniform in log scale.
   b. For the linear F study, set F_{size}=1.
2. Initialize $N = 3$ Moment-EFNs (with different seeds), where the F and ϕ networks have three layers of the above size and latent dimension L.
   a. For the linear F study, instead choose the F network to be a single linear layer.
3. Train all N Moment-EFNs, using BCE loss, with the same hyperparameters are the angularities study. Record their AUCs.
4. Record the mean and standard deviation of the N AUCs and plot a single point. Repeat for 25 total points.

Quark/Gluon Discrimination: Latents