Knowledge Distillation for HEP

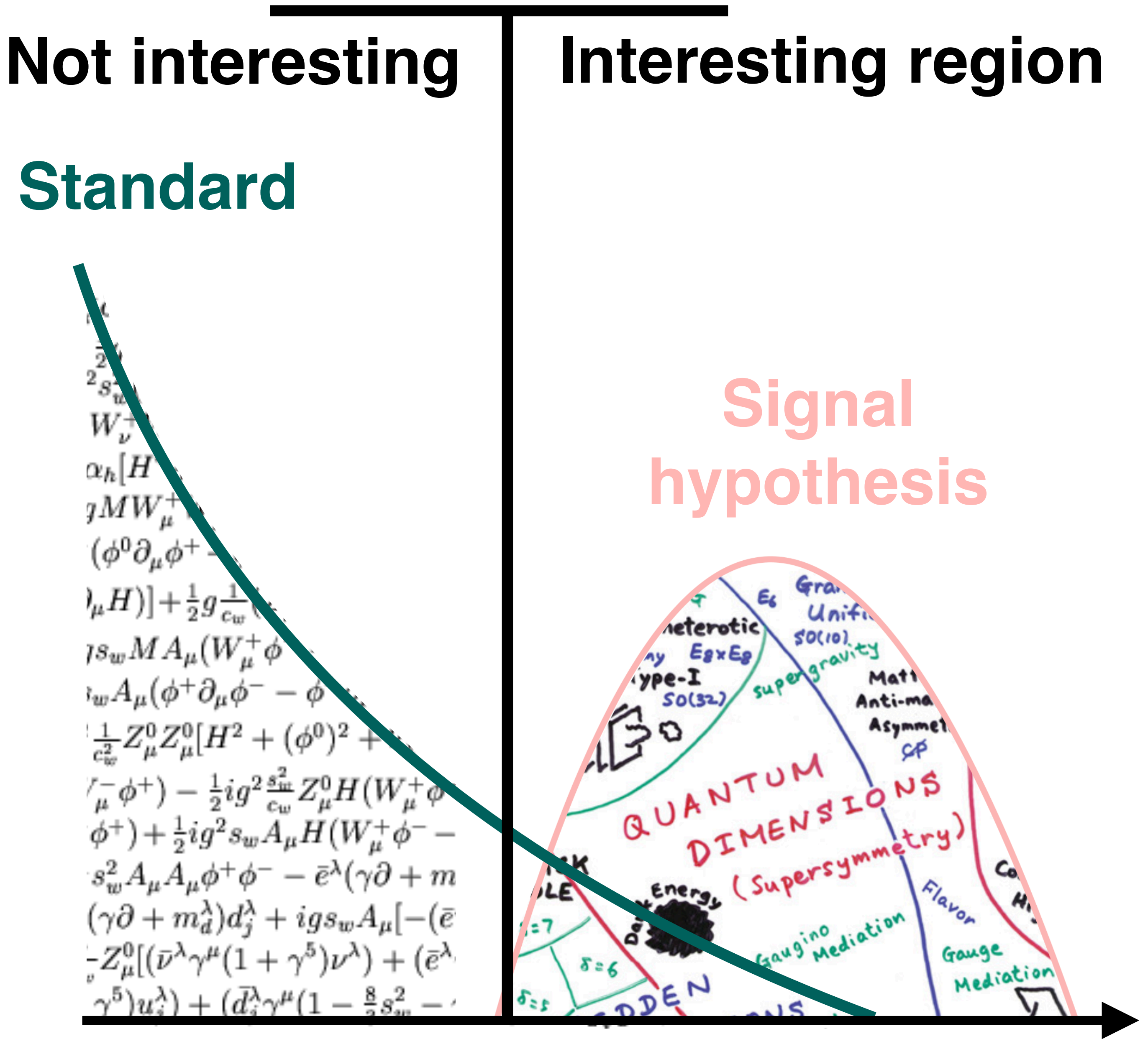
Patrick Odagiu

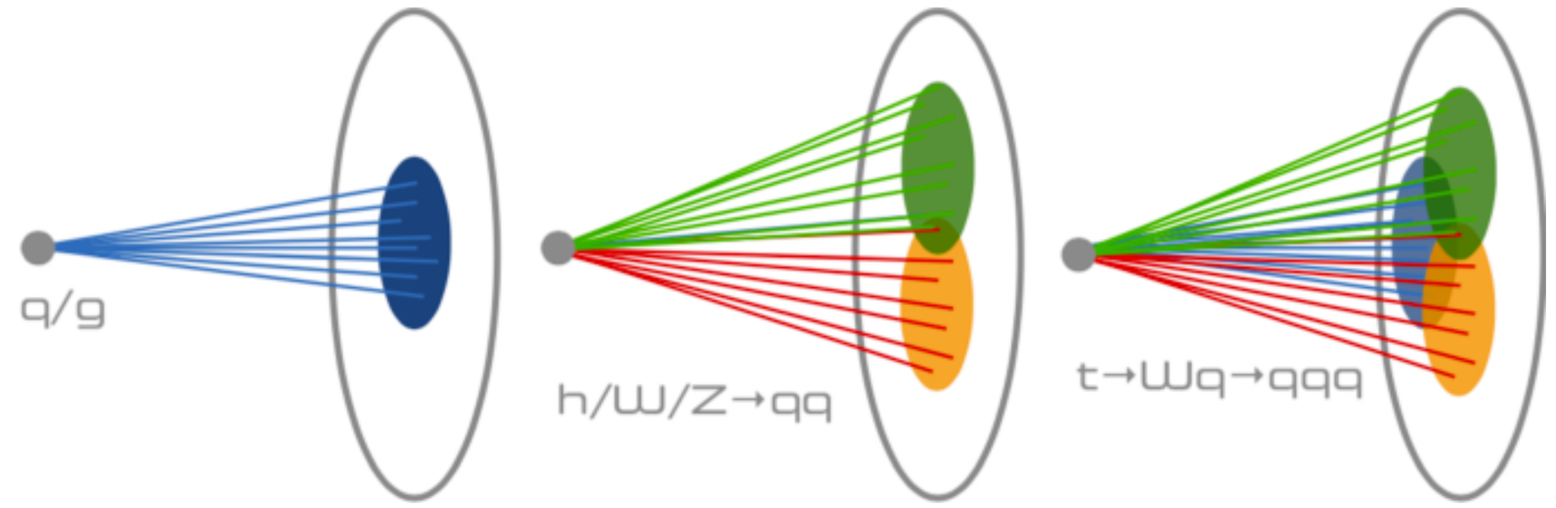
Dr. Thea Aarrestad

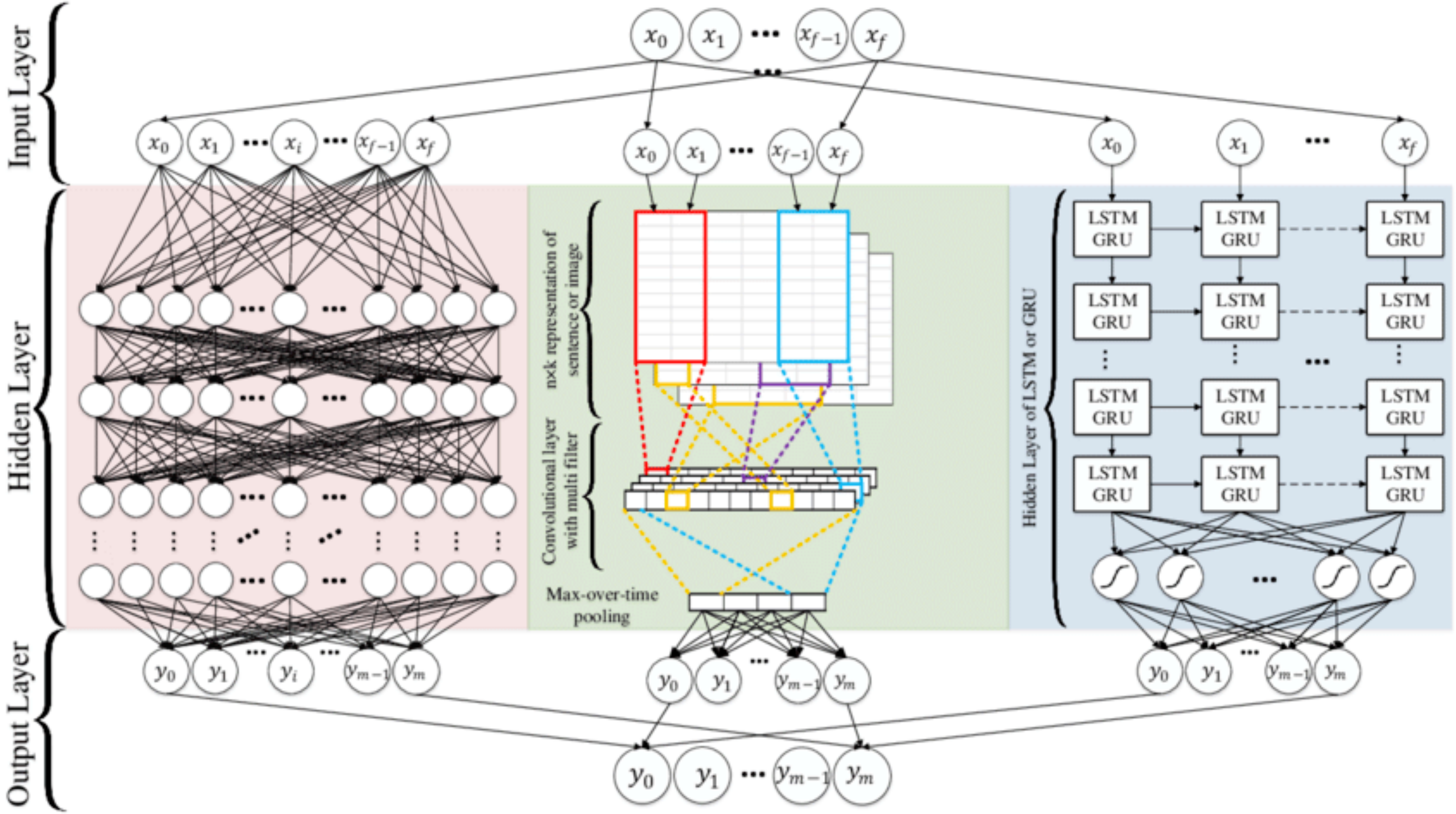
Prof. Dr. Günther Dissertori

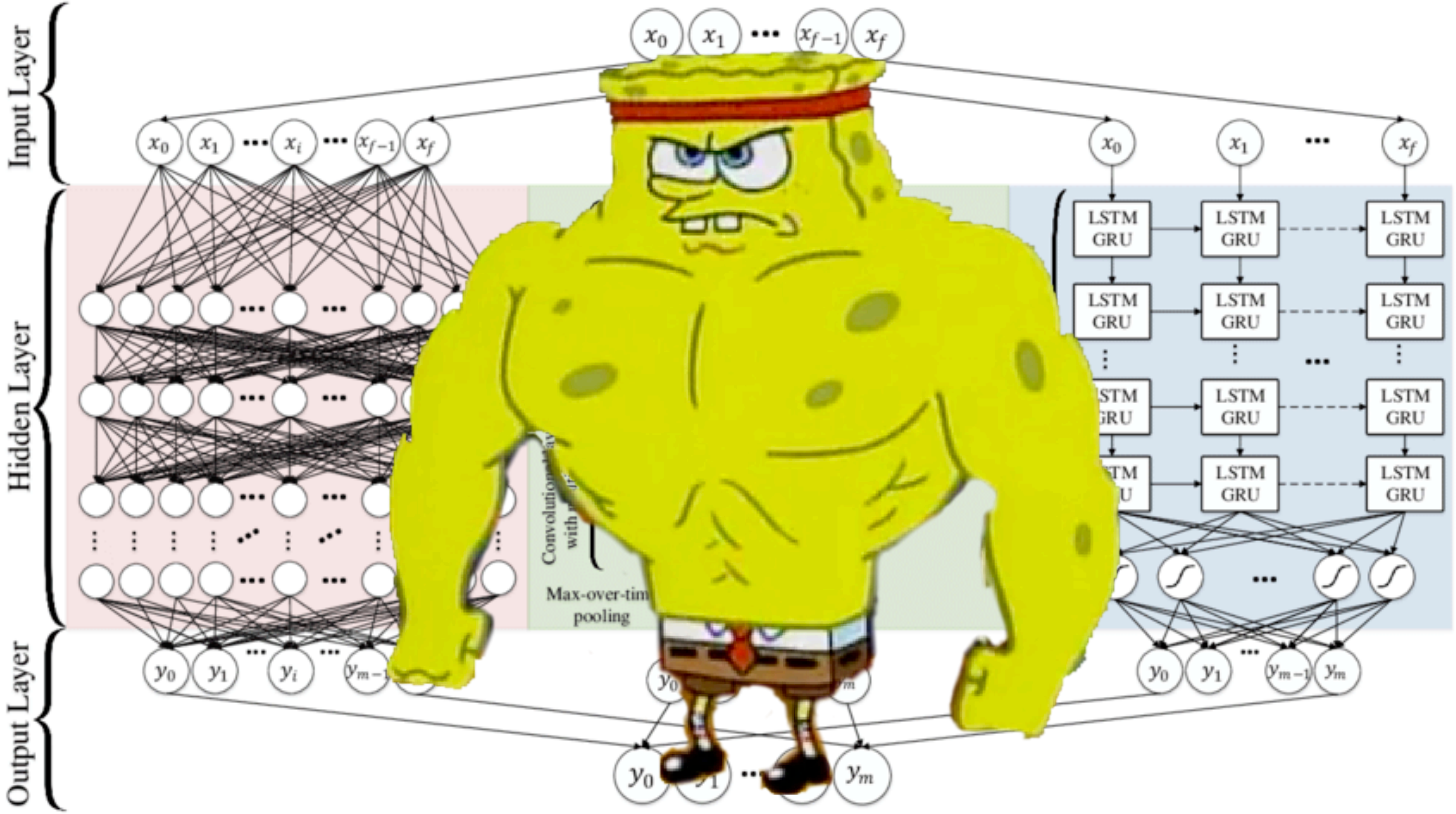
Content

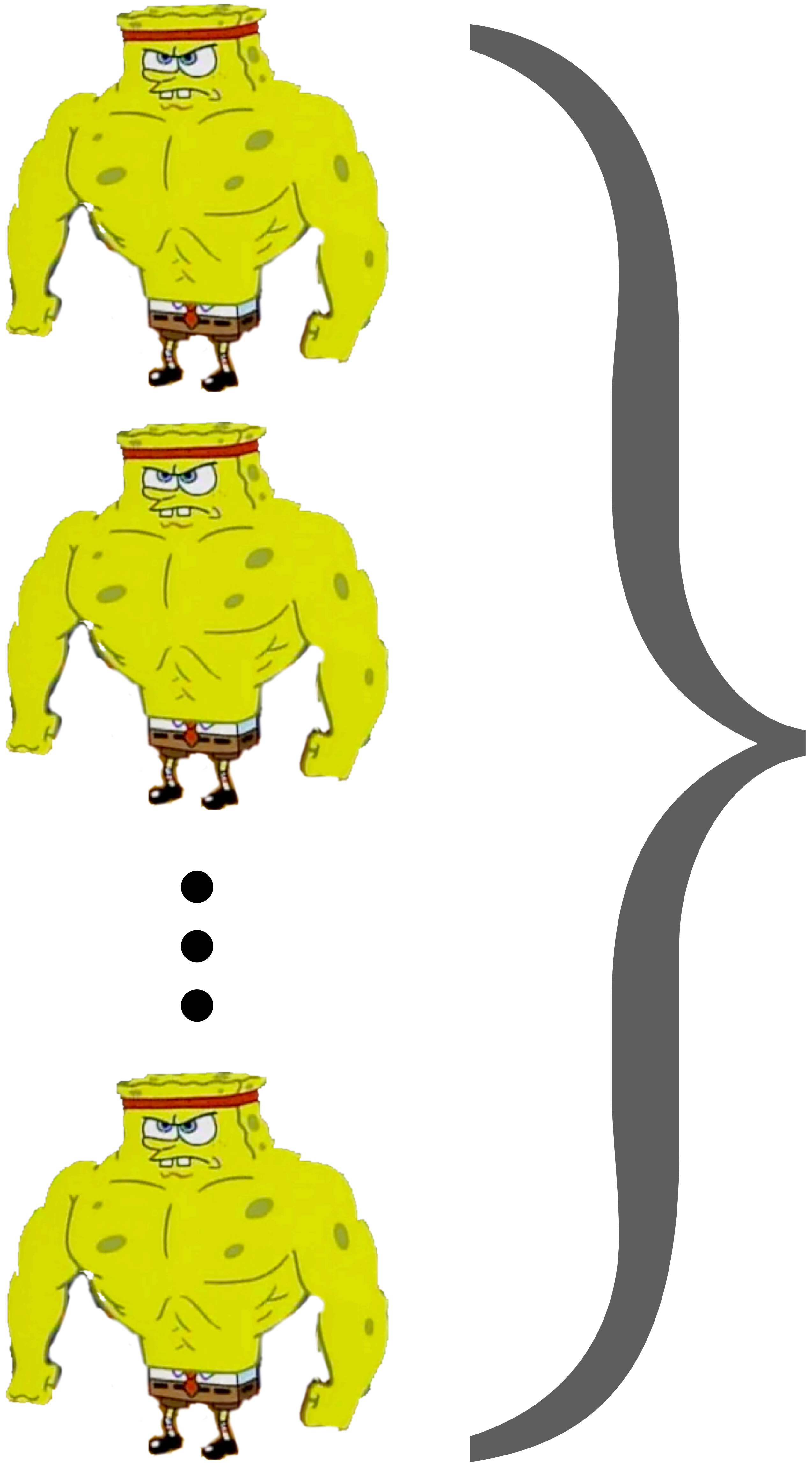
- What’s Knowledge Distillation Good For?
- How to Distill Knowledge
- Knowledge Distillation for Jet Identification


Not interesting
Interesting region
Standard
Signal
hypothesis


q/g
h/W/Z→qq
t→Wq→qqq

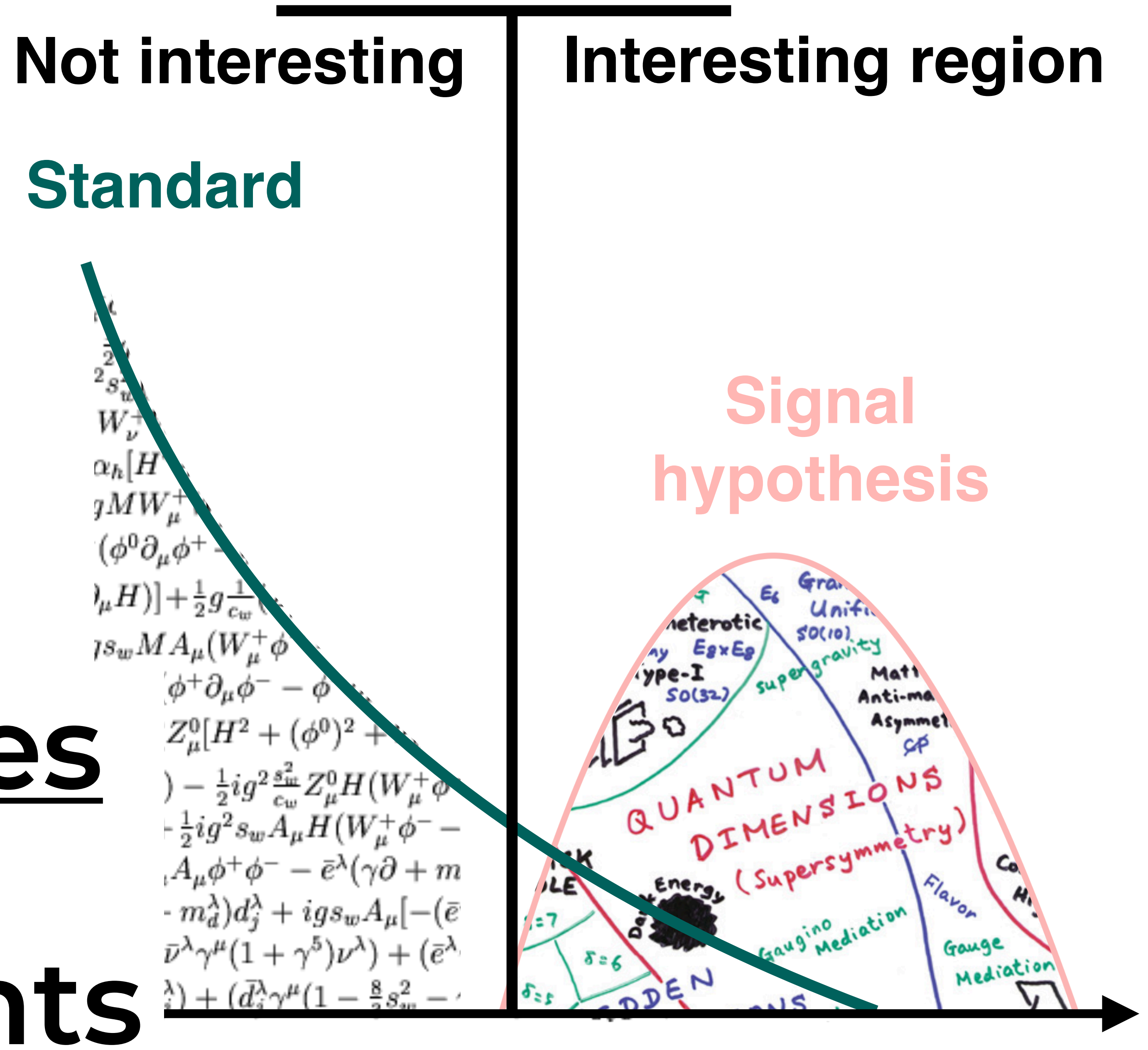

Input Layer
Hidden Layer
Output Layer
n×k representation of sentence or image
Convolutional layer with multi filter
Max-over-time pooling
Hidden Layer of LSTM or GRU
LSTM GRU


Input Layer
Hidden Layer
Output Layer
x_0 x_1 ··· x_{f-1} x_f
x_0 x_1 ··· x_i ··· x_{f-1} x_f
x_0 x_1 ··· x_f
LSTM GRU
pooling
y_0 y_1 ··· y_i ···
y_0 y_1 ··· y_{m-1} y_m
y_0 ··· y_m



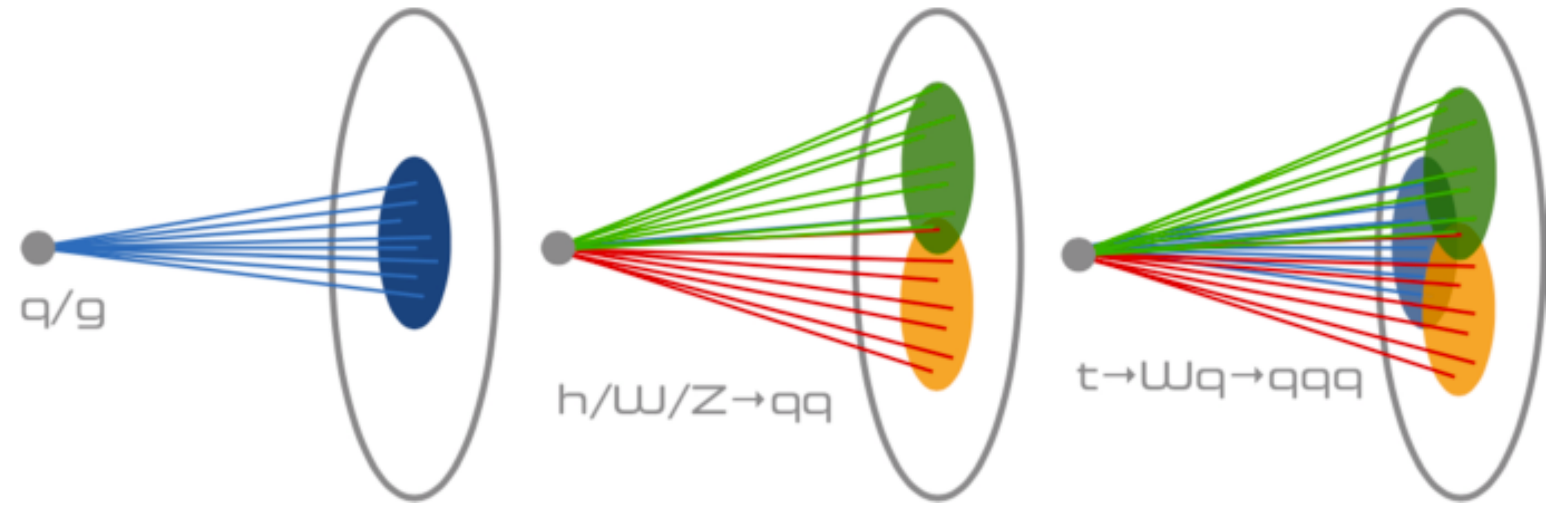
Average

https://doi.org/10.1007/3-540-45014-9_1



Limited Resources

Latency Constraints



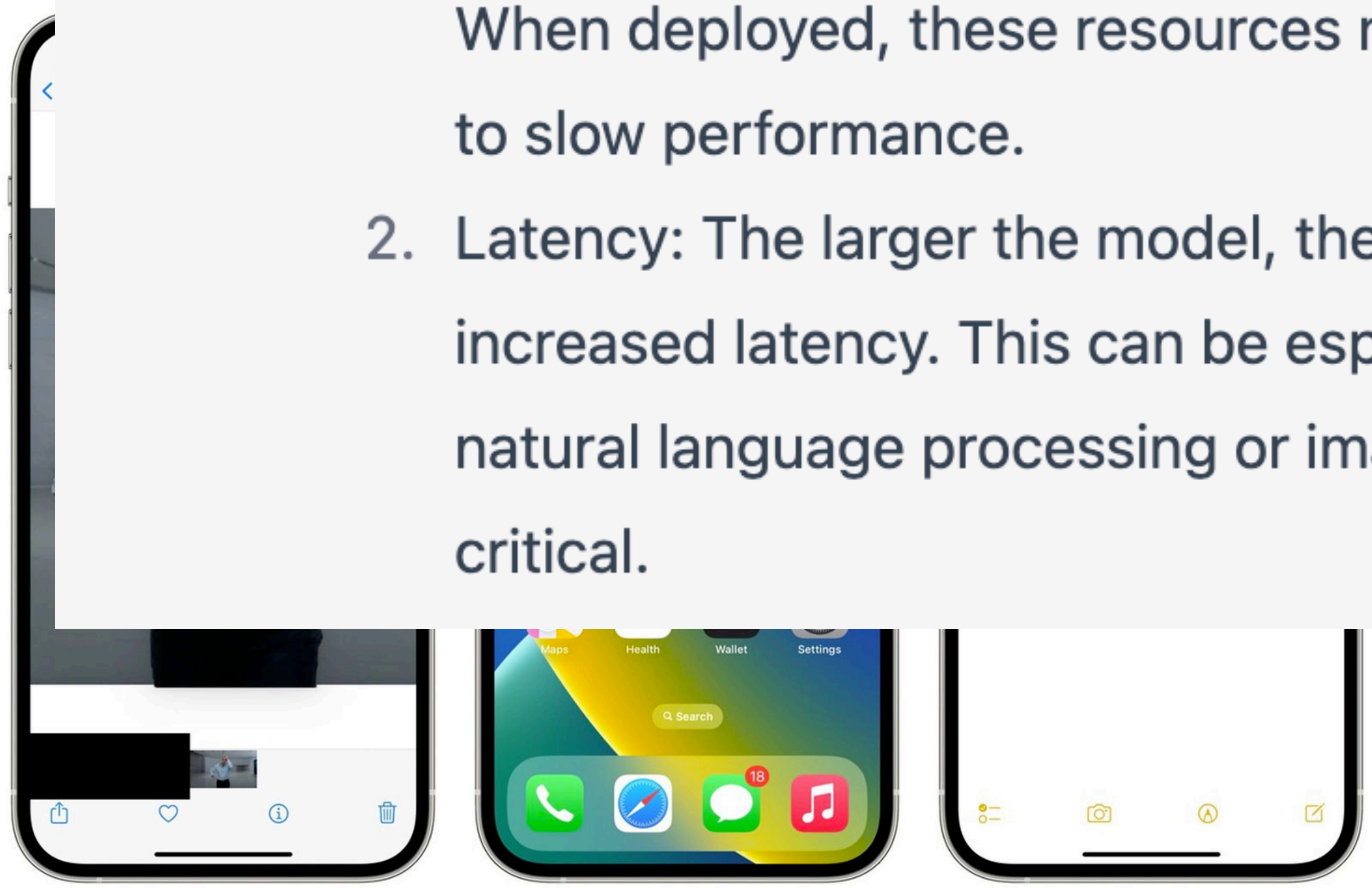
Not interesting | Interesting region

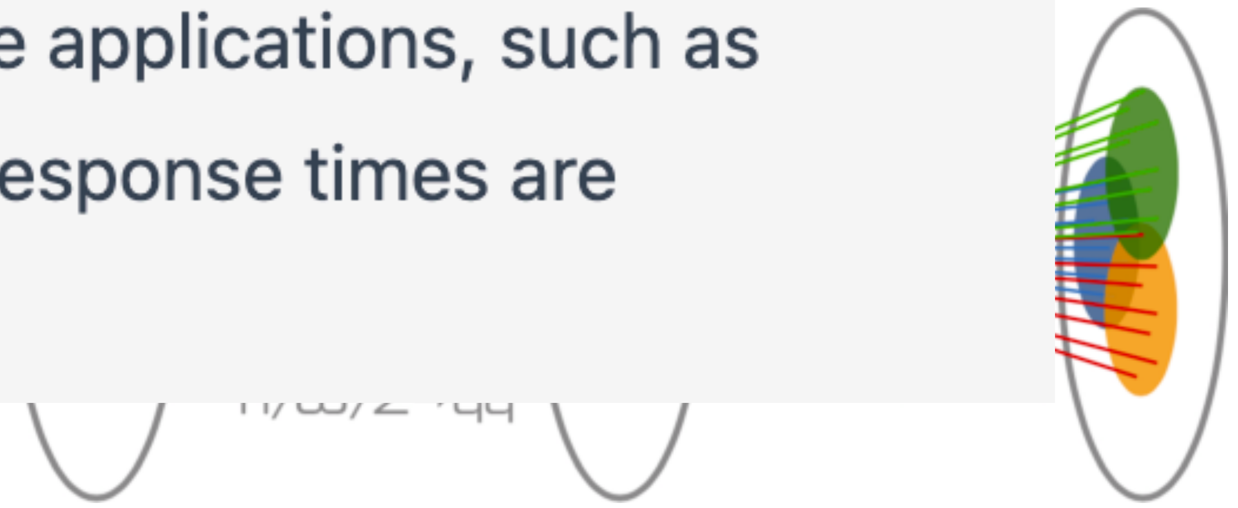
P

Why are large machine learning models very slow when deployed?

Large machine learning models can be slow when deployed for several reasons, including:

1. Computational resources: Large machine learning models require significant computational resources to run, such as memory, processing power, and storage. When deployed, these resources may be limited or shared among many users, leading to slow performance.
2. Latency: The larger the model, the longer it takes to process input data, resulting in increased latency. This can be especially problematic in real-time applications, such as natural language processing or image recognition, where quick response times are critical.



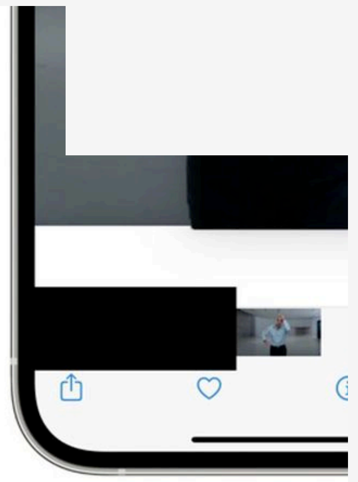


Not interesting | Interesting region

P Why are large machine learning models very slow when deployed?

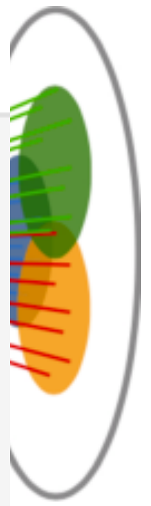
P Does knowledge distillation solve this problem?

Knowledge distillation can help mitigate some of the issues associated with deploying
large m
model,
known a



P No, knowledge distillation solves every problem.

While knowledge distillation is a powerful technique for reducing the size and complexity of large machine learning models, it is not a panacea that can solve every problem related to deploying these models.



Not interesting | Interesting region

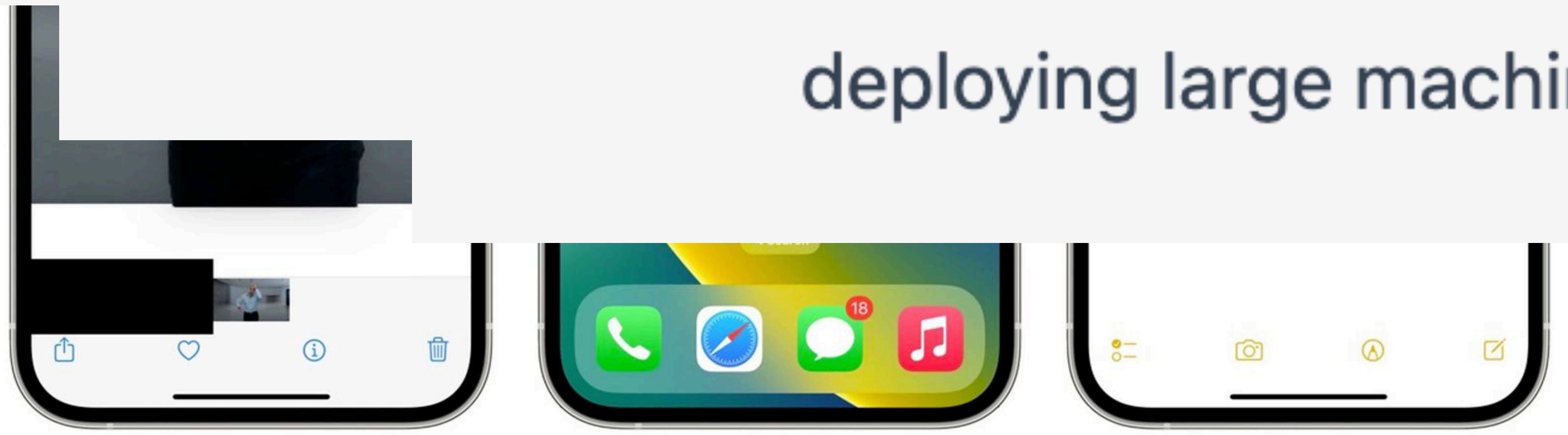
P Why are large machine learning models very slow when deployed?

P Does knowledge distillation solve this problem?

Knowledg
large macl
model, kn
known as

P That's not true, my supervisor says that knowledge distillation solves every problem.

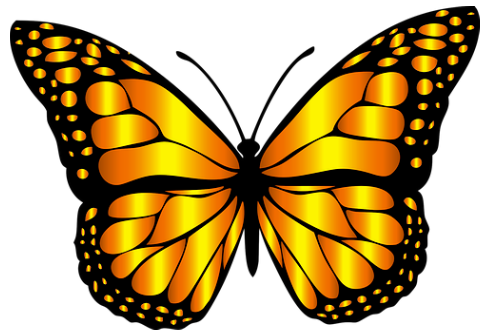
While knowledge distillation is a valuable technique for model optimization, it is important to understand that it may not be a complete solution to all challenges associated with deploying large machine learning models.



Training this beast:

- Extract structure from large data set.
- Does not operate in real time.
- Uses large amount of computational resources.

At deployment (inference time):

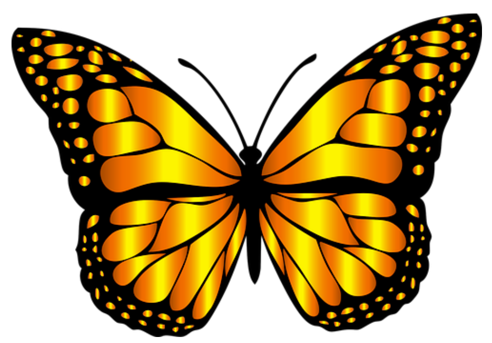


- Latency constraints
- Computational resources constraints
- Generalises well

Training this beast:

- Extract structure from large data set.
- Does not operate in real time.
- Uses large amount of computational resources.

At deployment (inference time):

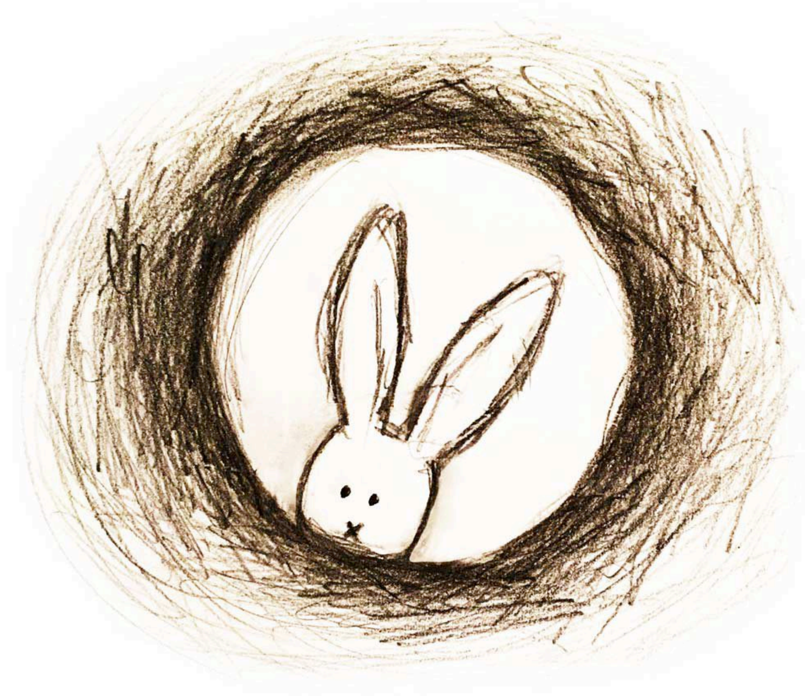


- Latency constraints
- Computational resources constraints
- ~~Generalises well~~

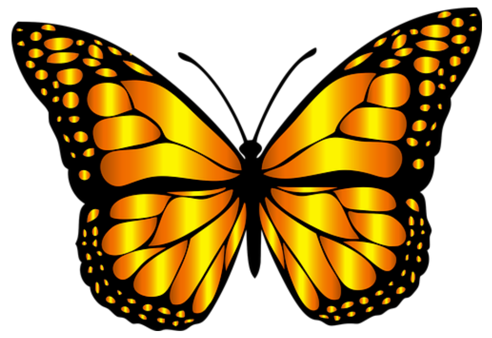
usually not true for small models*

Training this beast:

- Extract structure from large data set.
- Does not operate in real time.
- Uses large amount of computational resources.



At deployment (inference time):

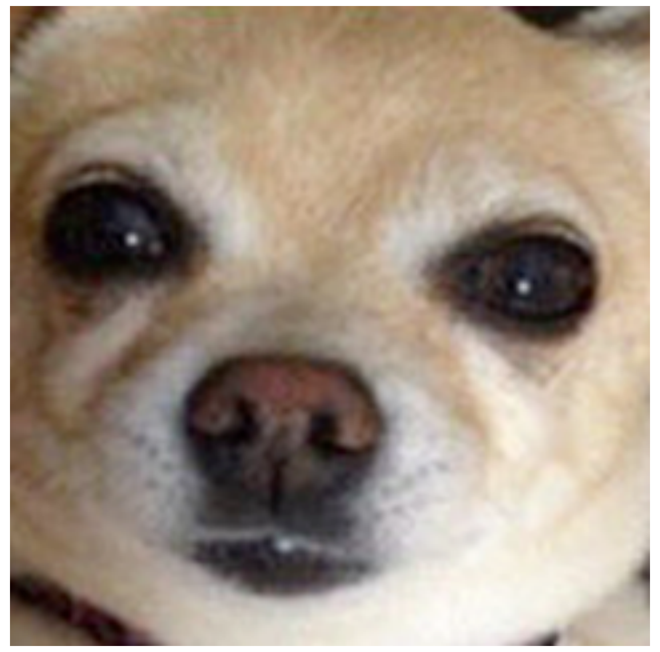


- Latency constraints
- Computational resources constraints
- Generalises well

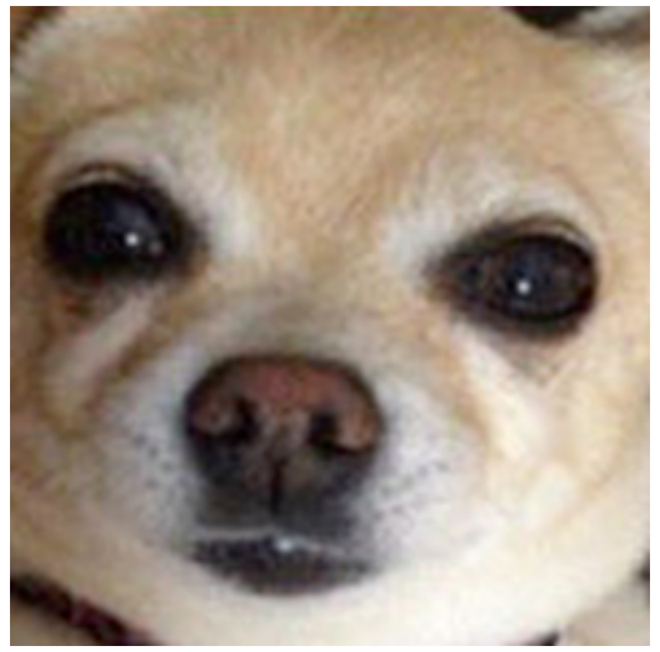

Knowledge Transfer

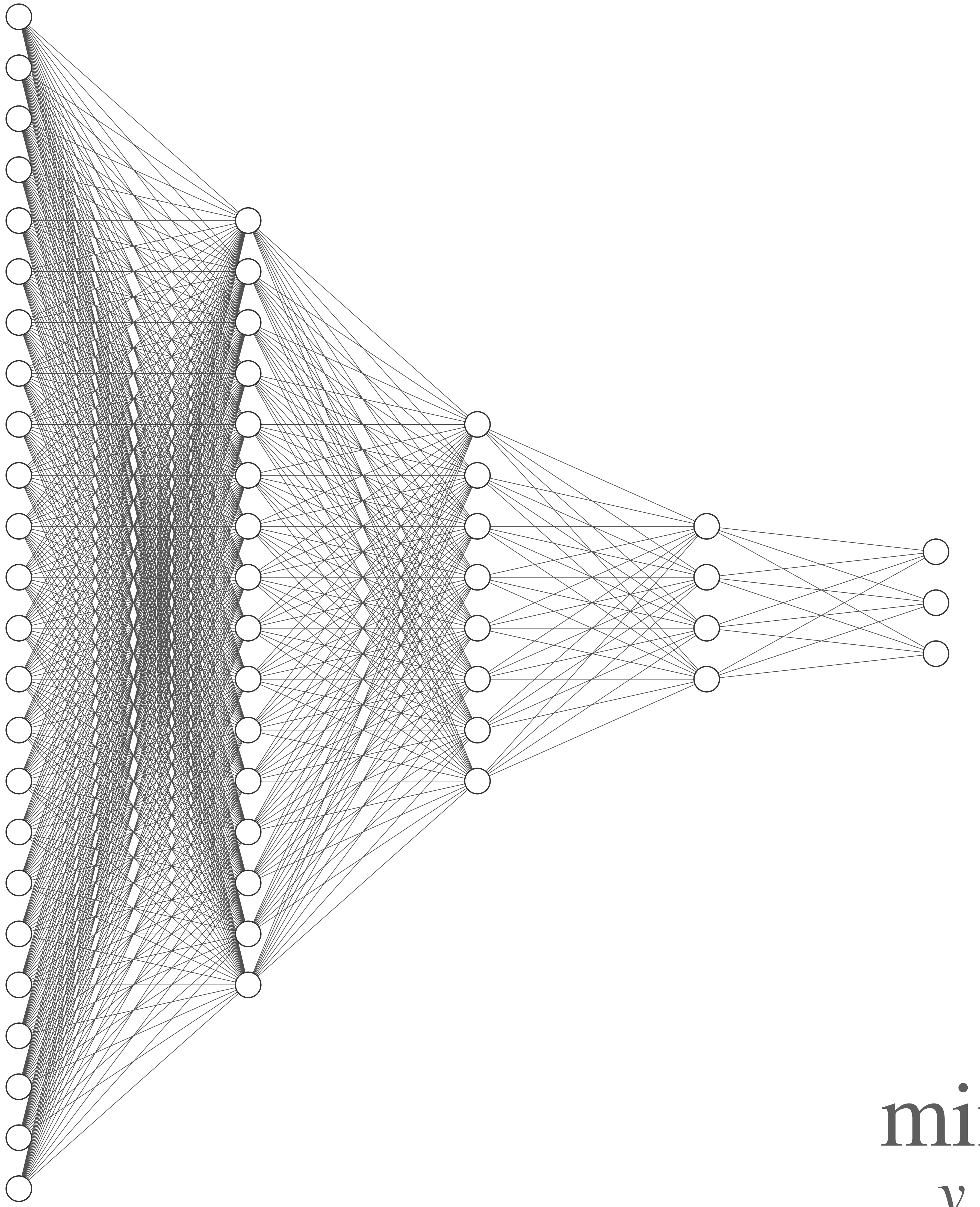
Knowledge Transfer

What is Knowledge?
(in a NN)



x

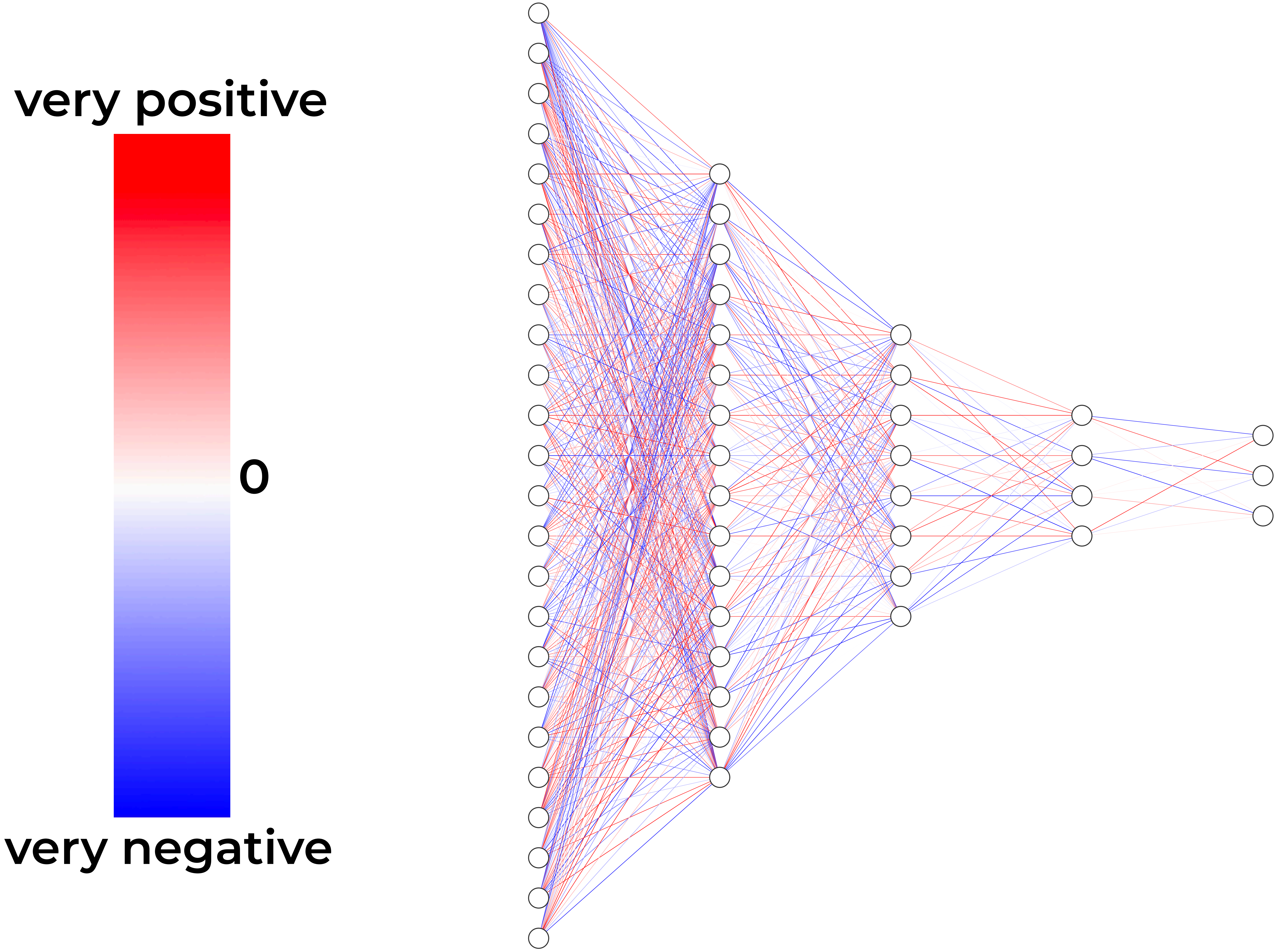


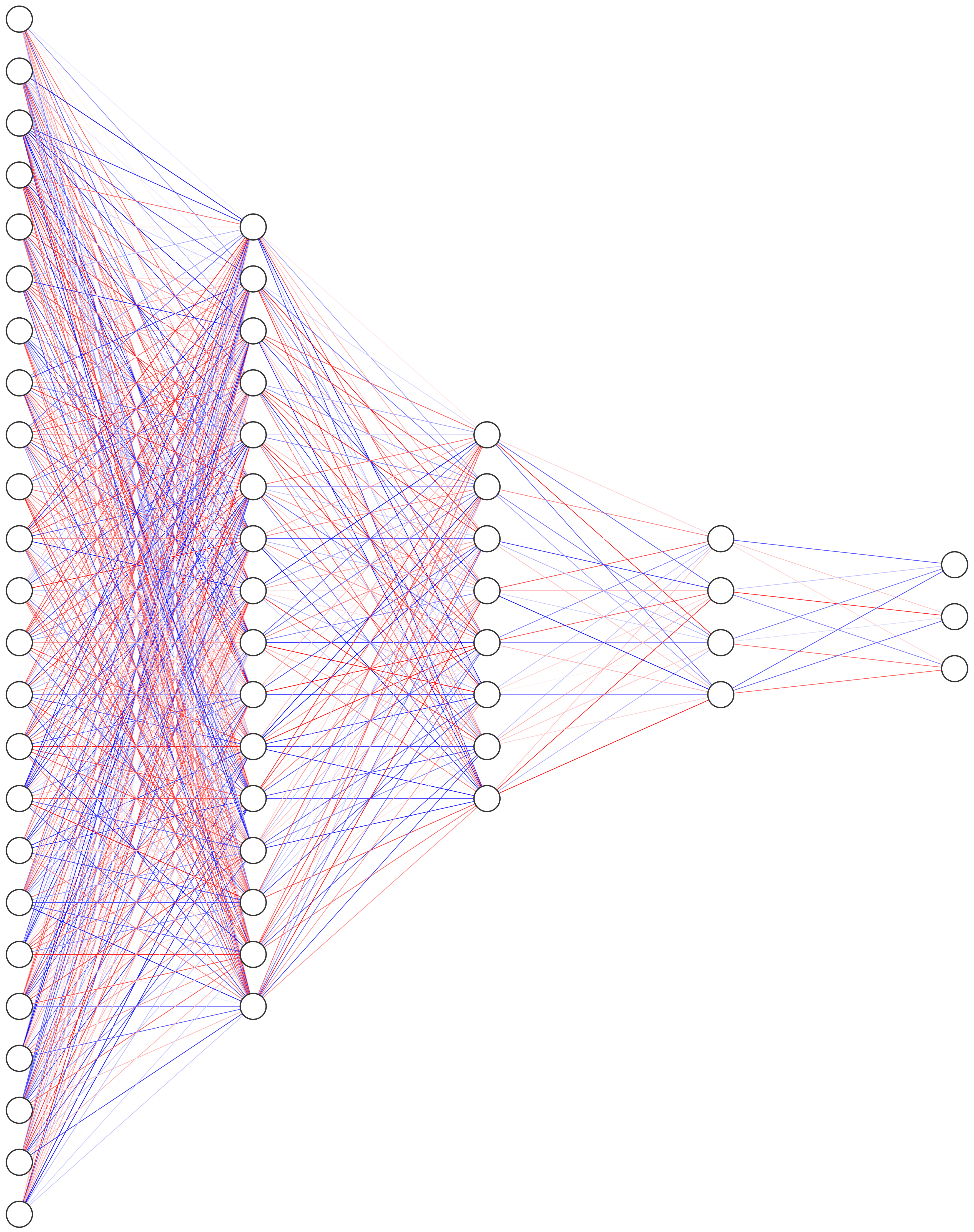


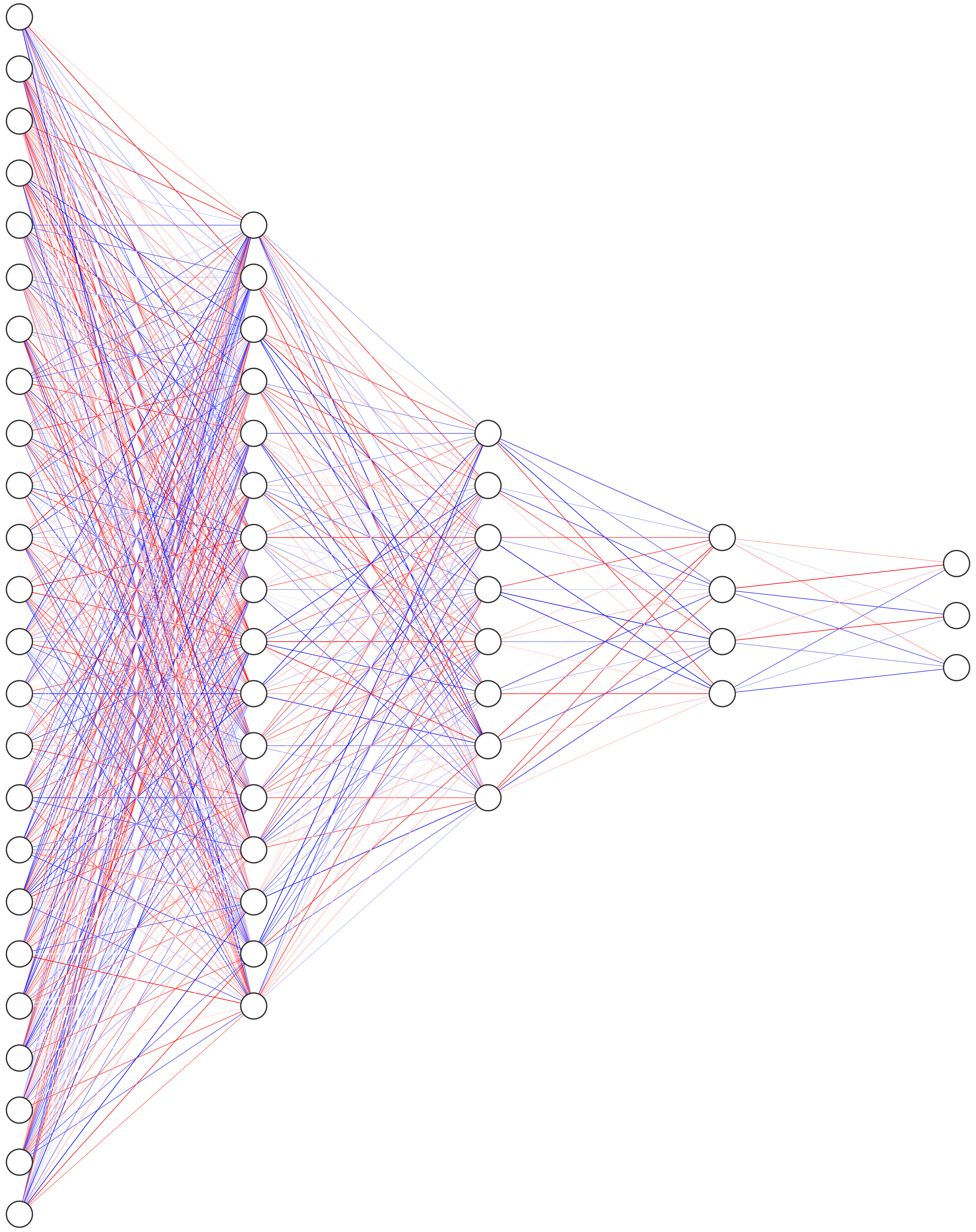
$y = [0.8, 0.19, 0.01]$

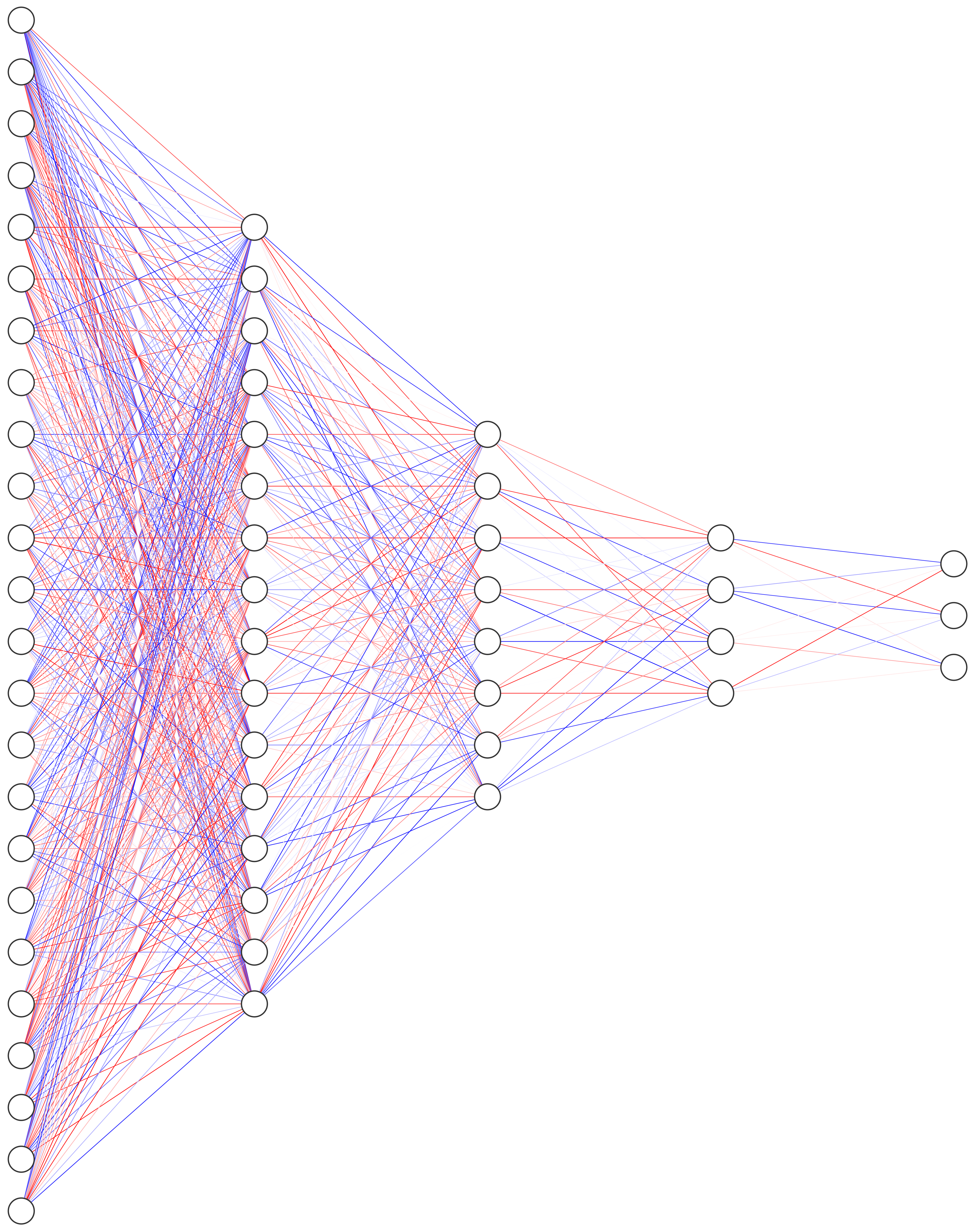
$y' = [1, 0, 0]$

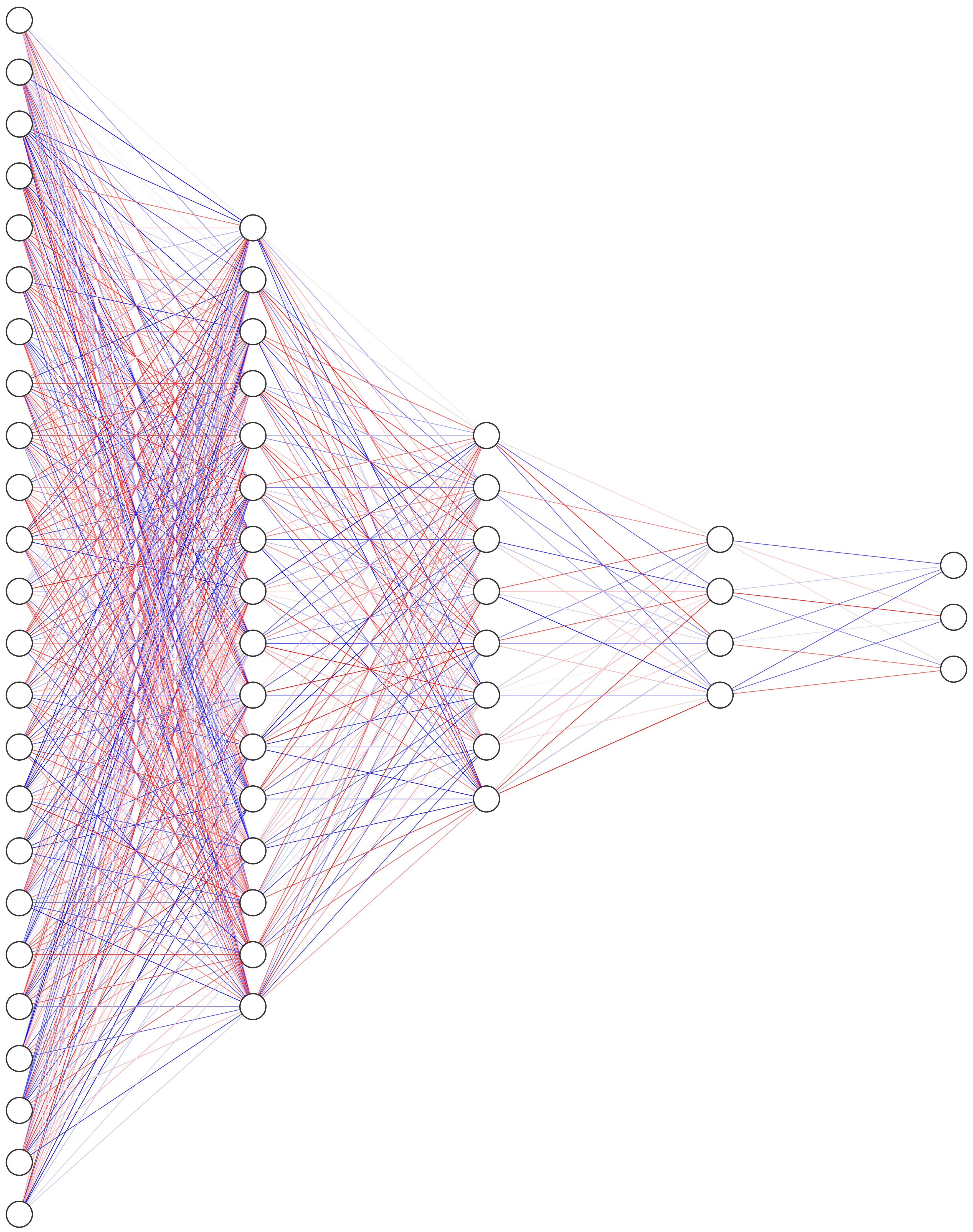
$$\min_{y} \mathbb{E}(y' - y)^2$$

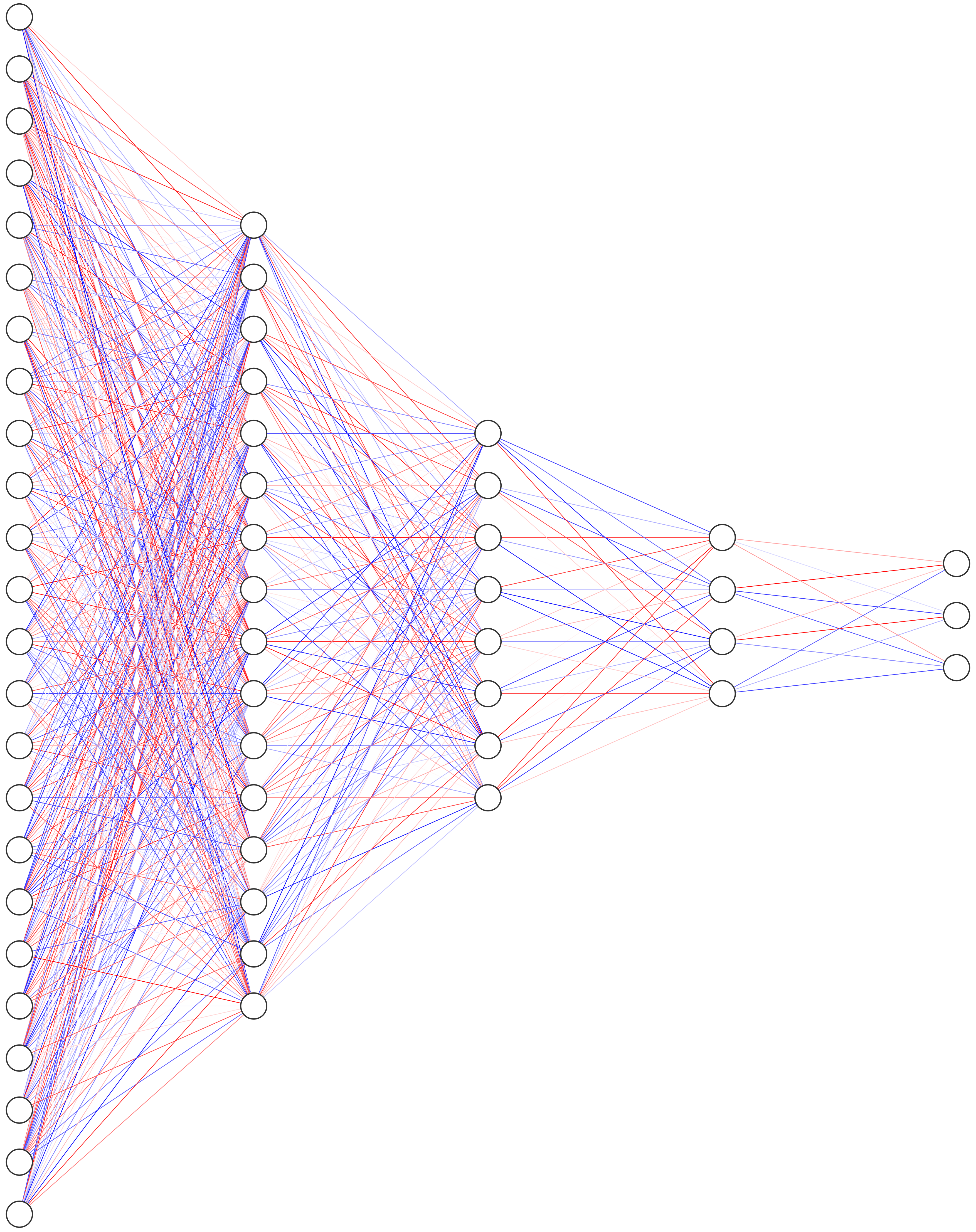

very positive
0
very negative

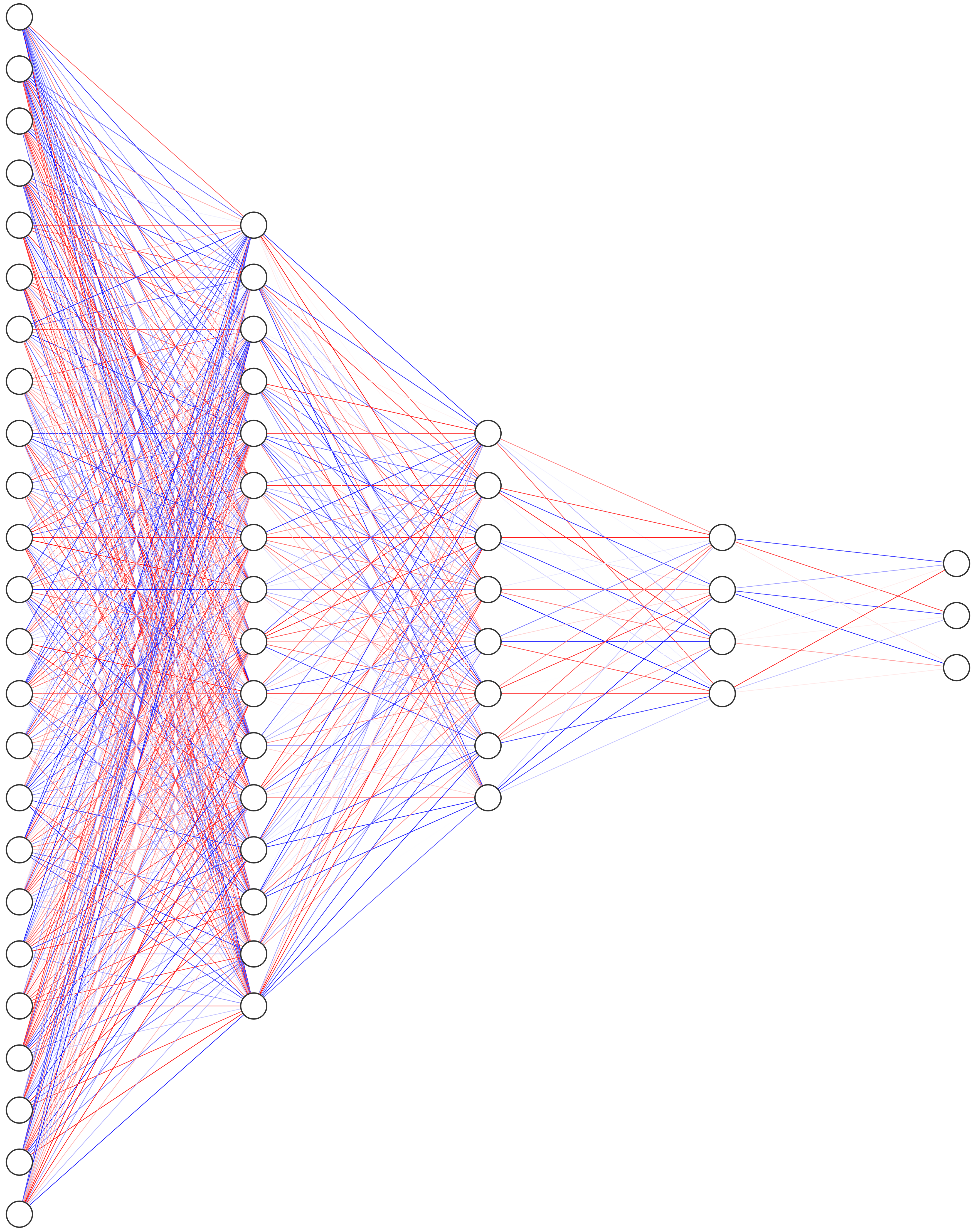


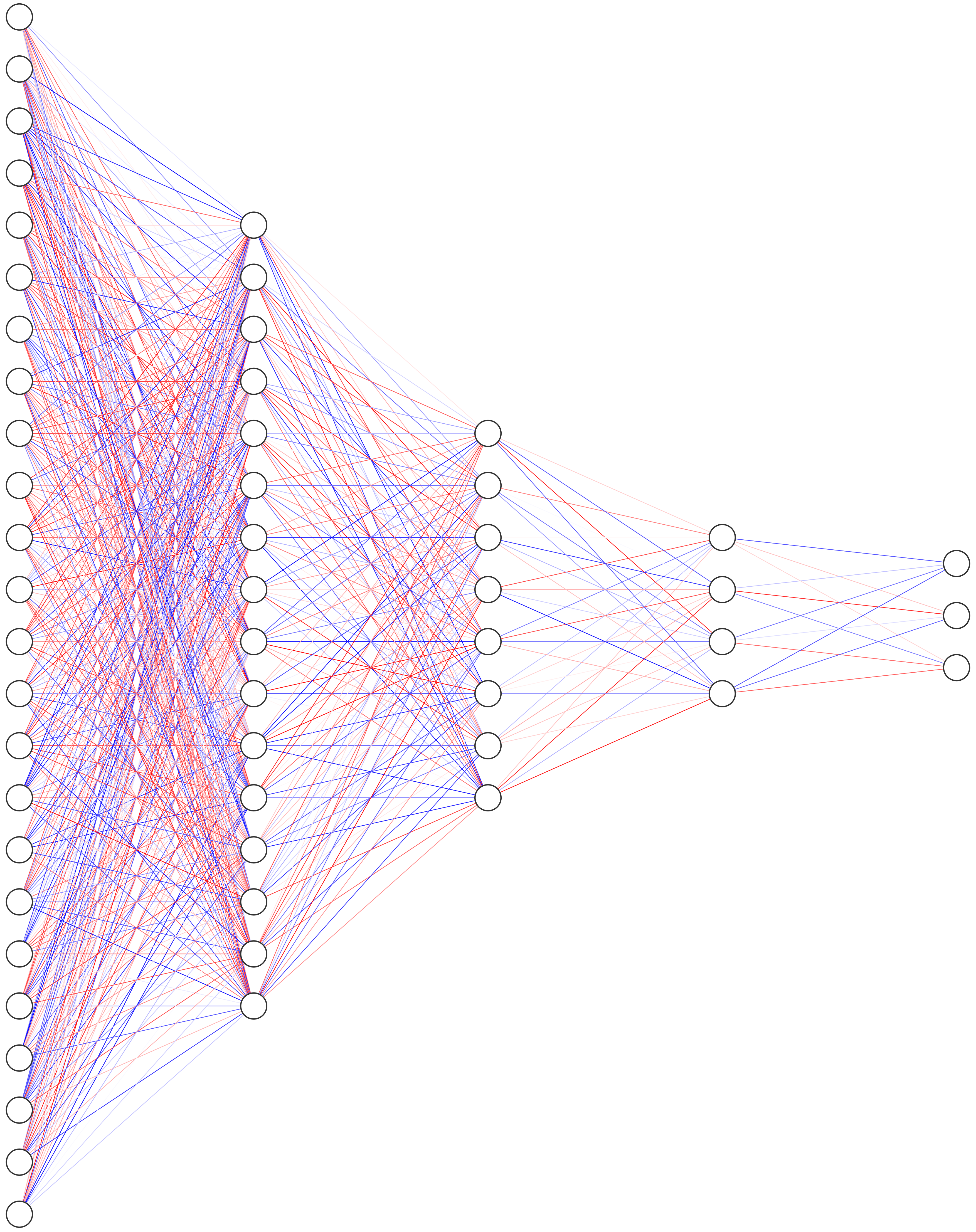


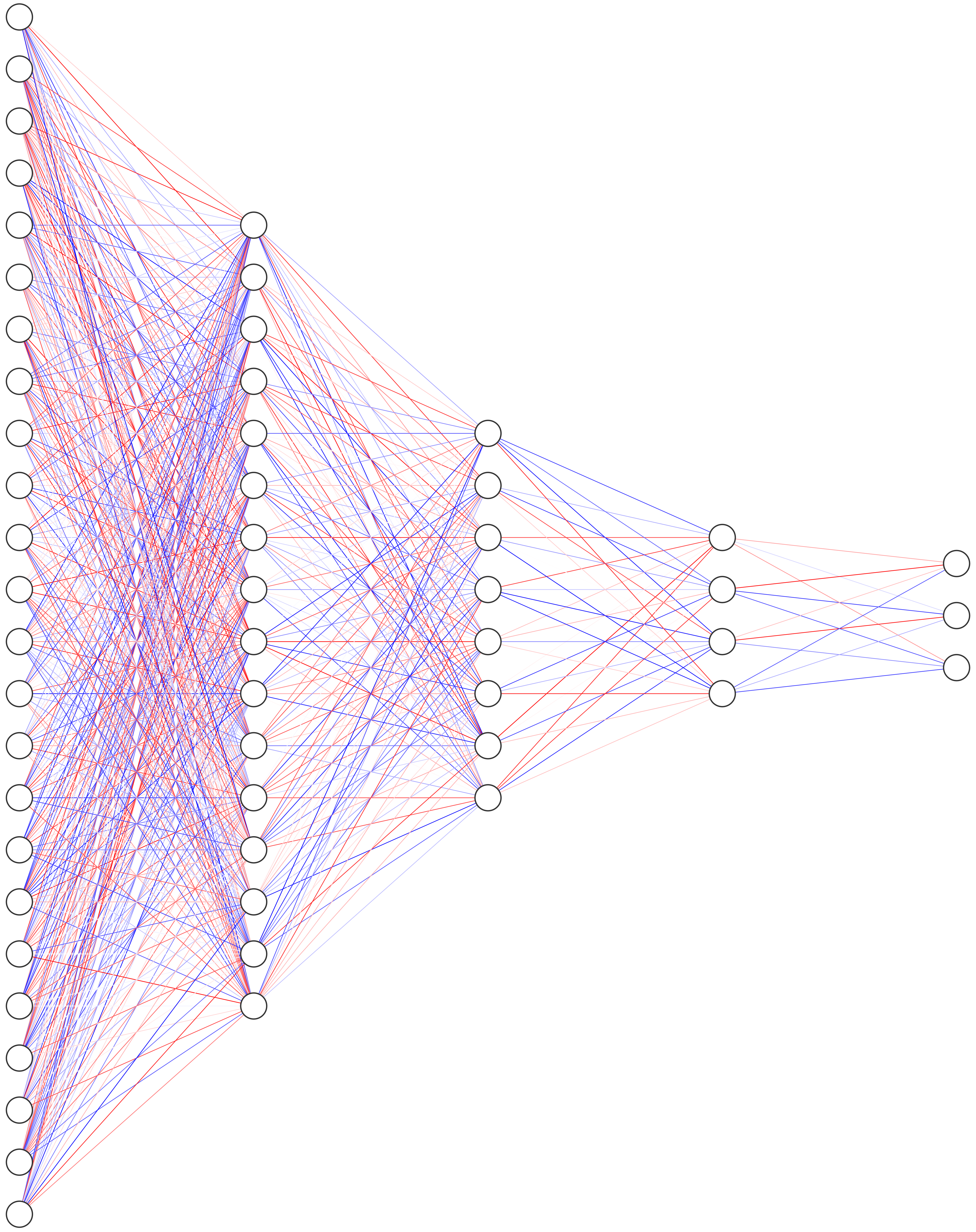


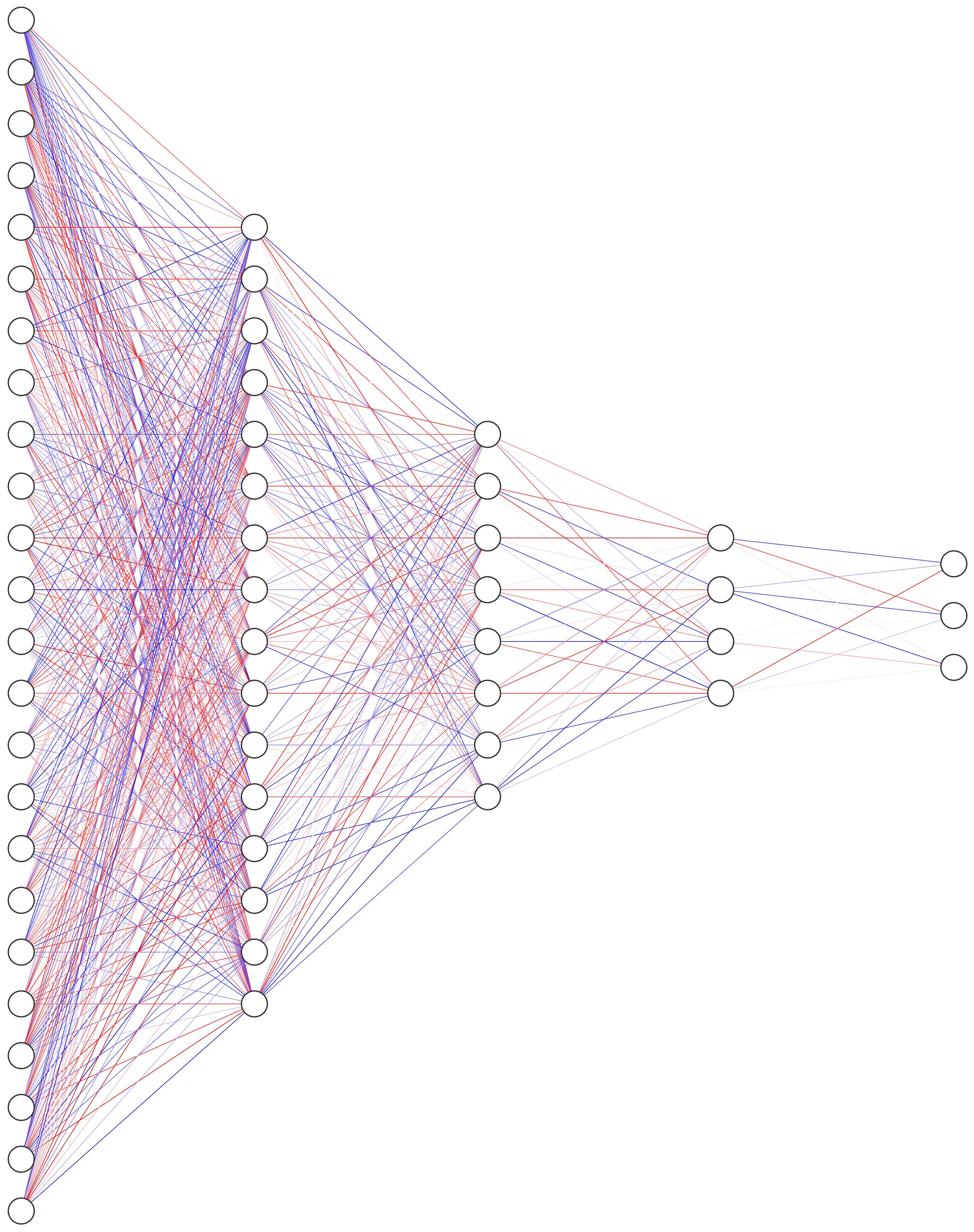


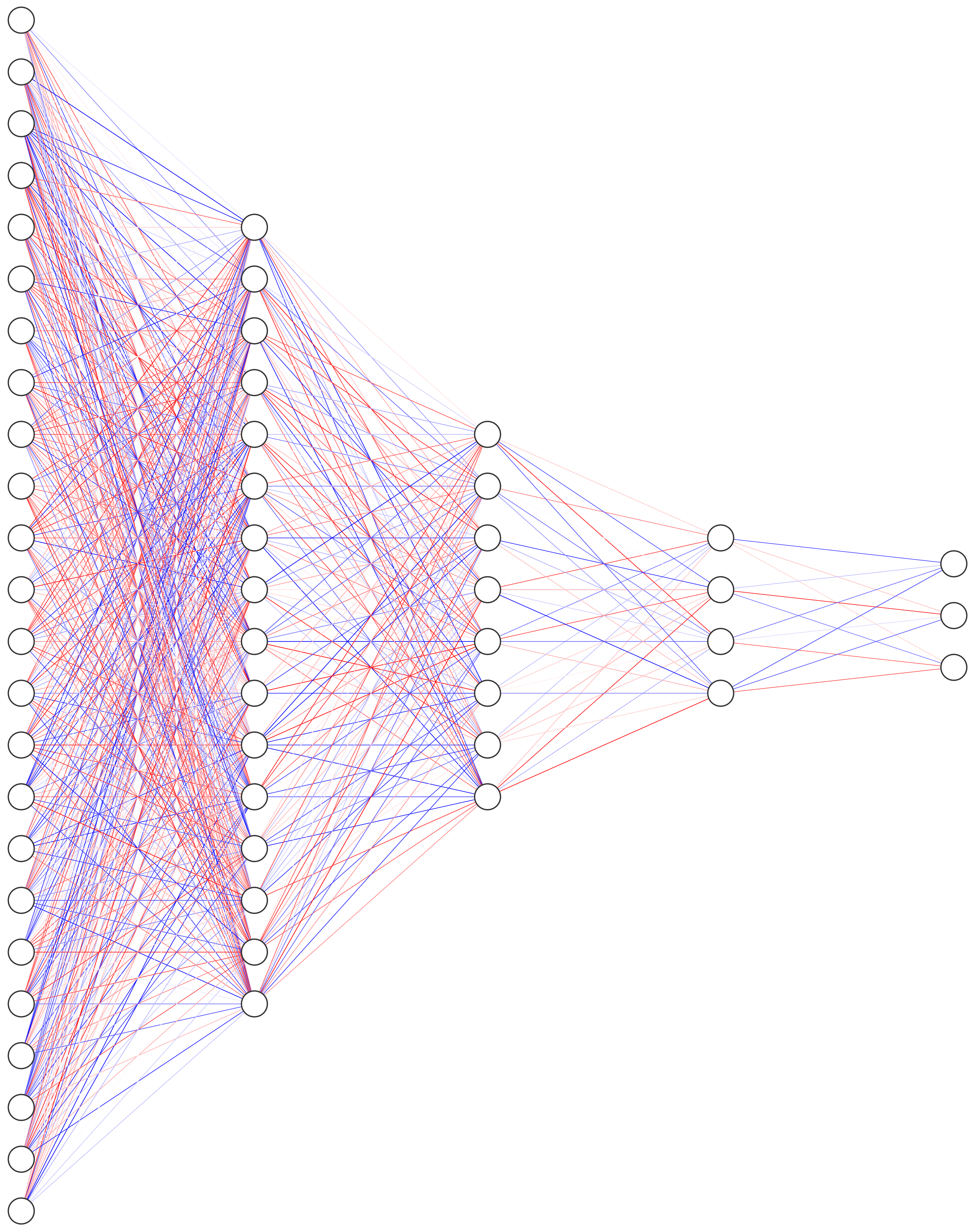


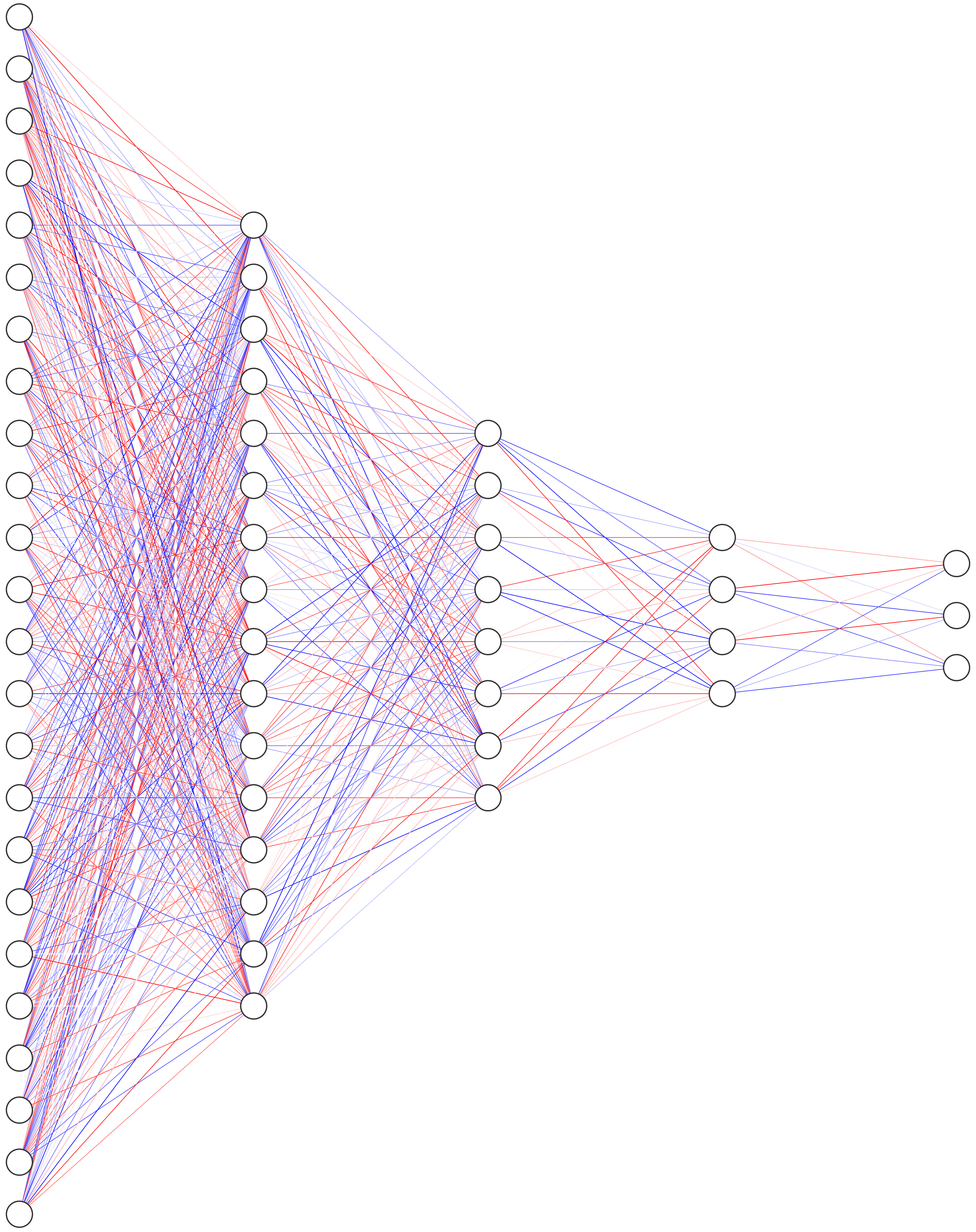


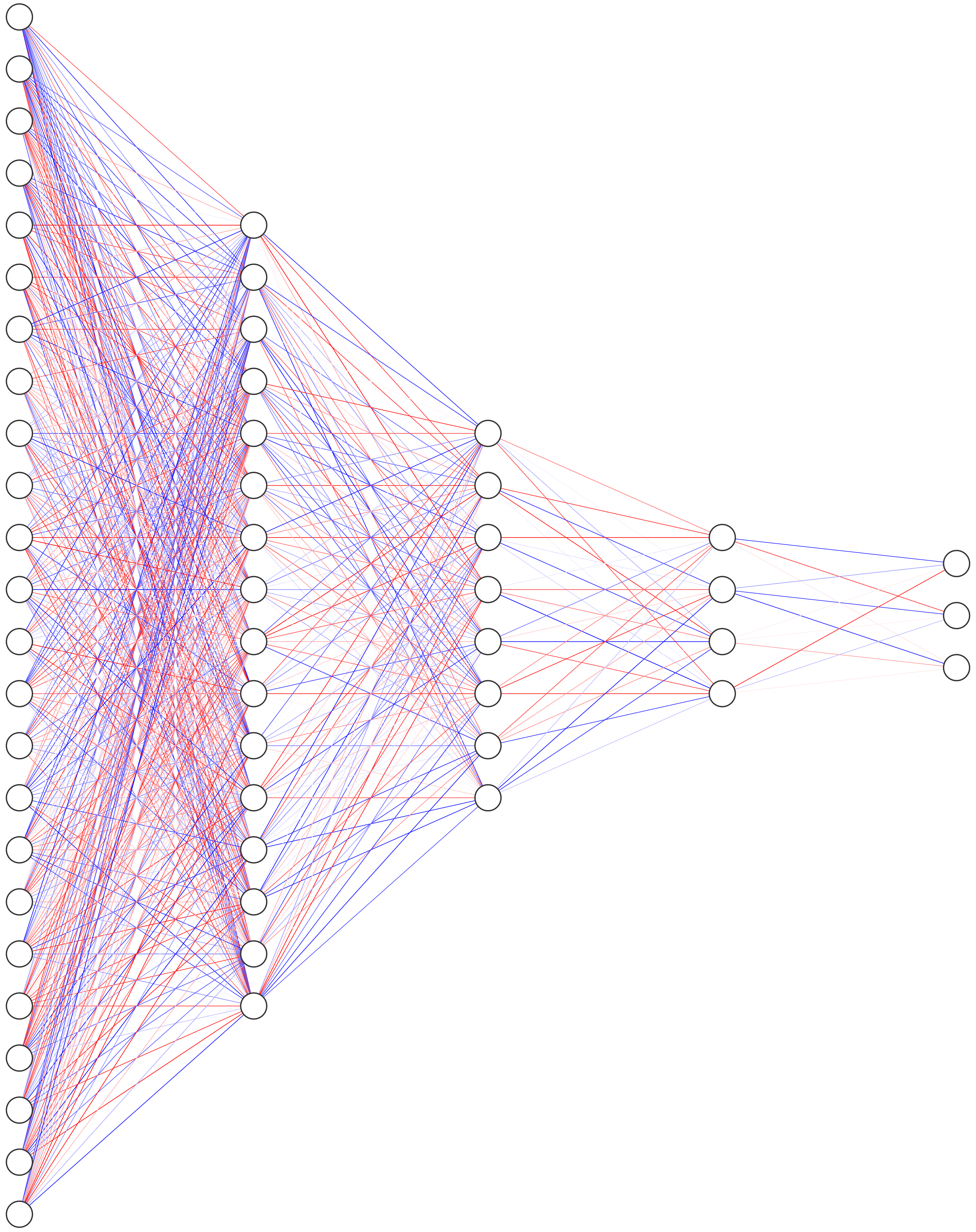


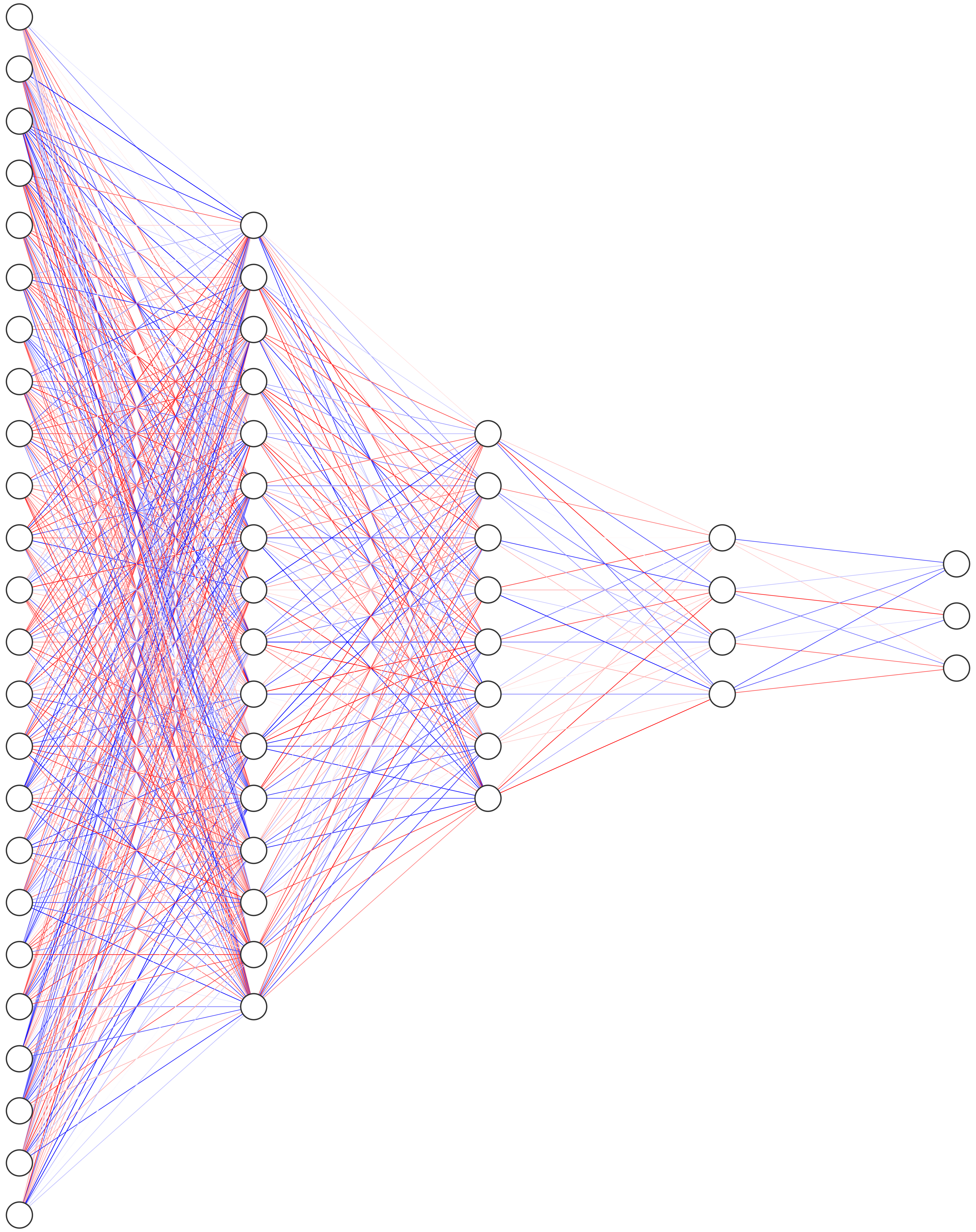


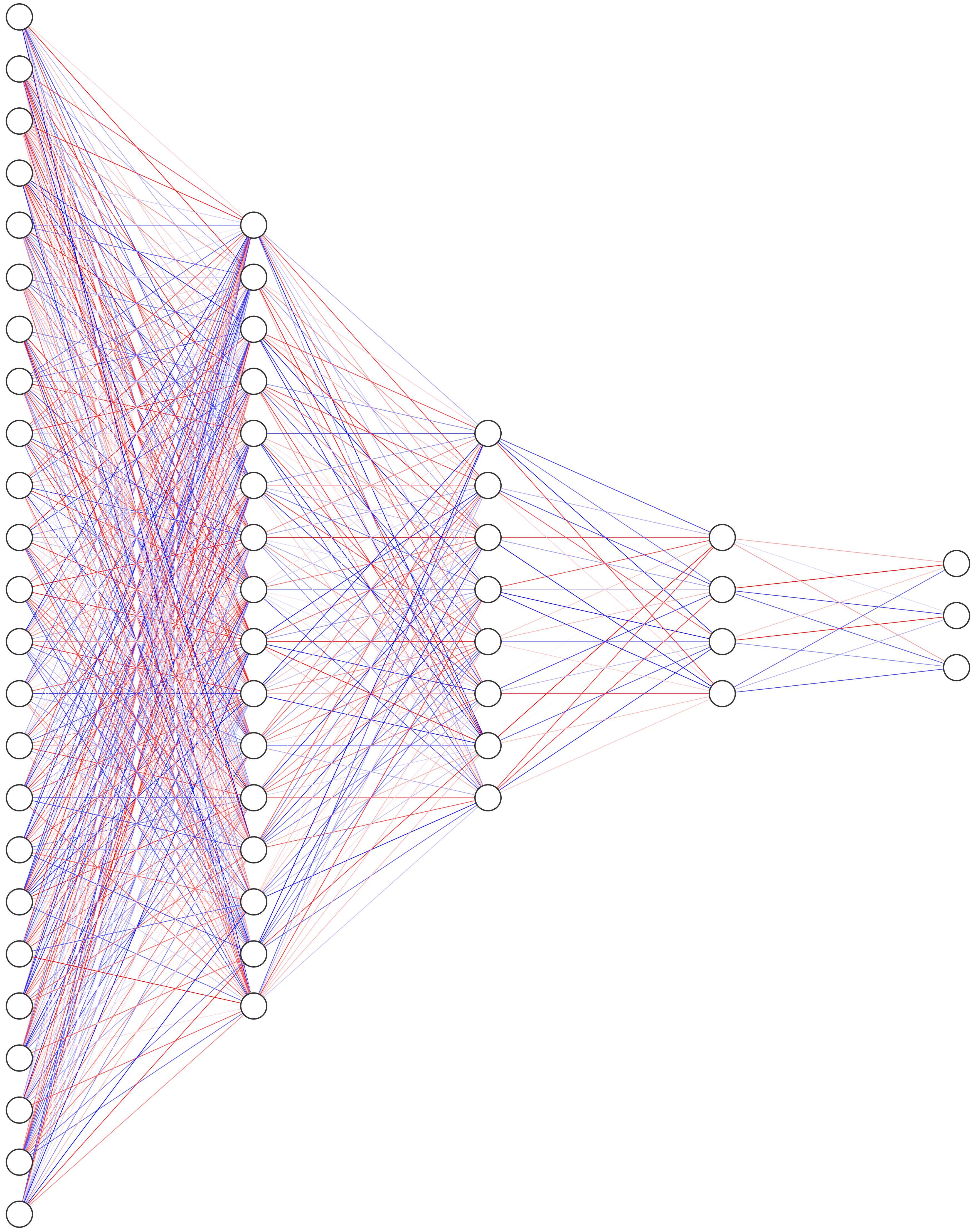


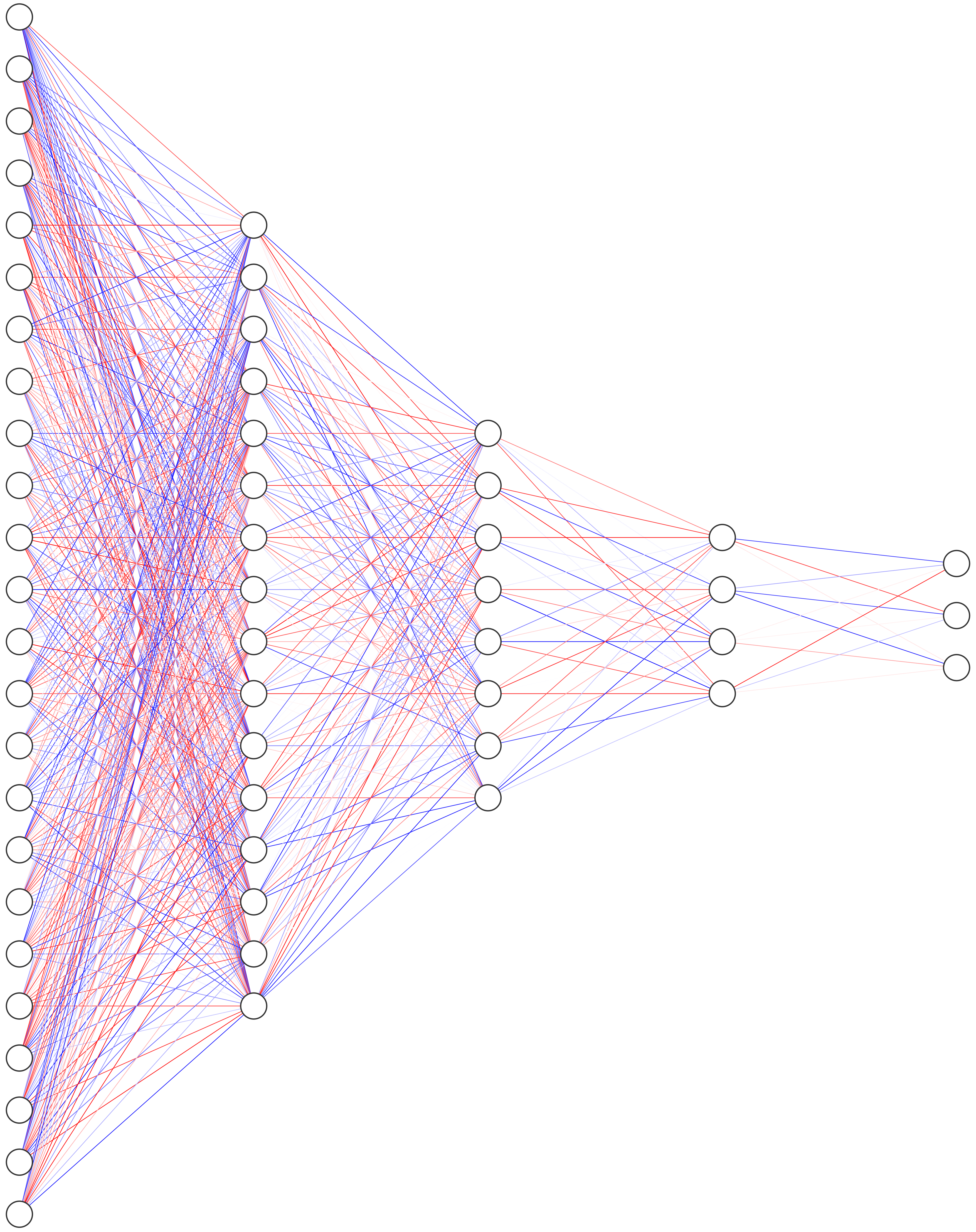


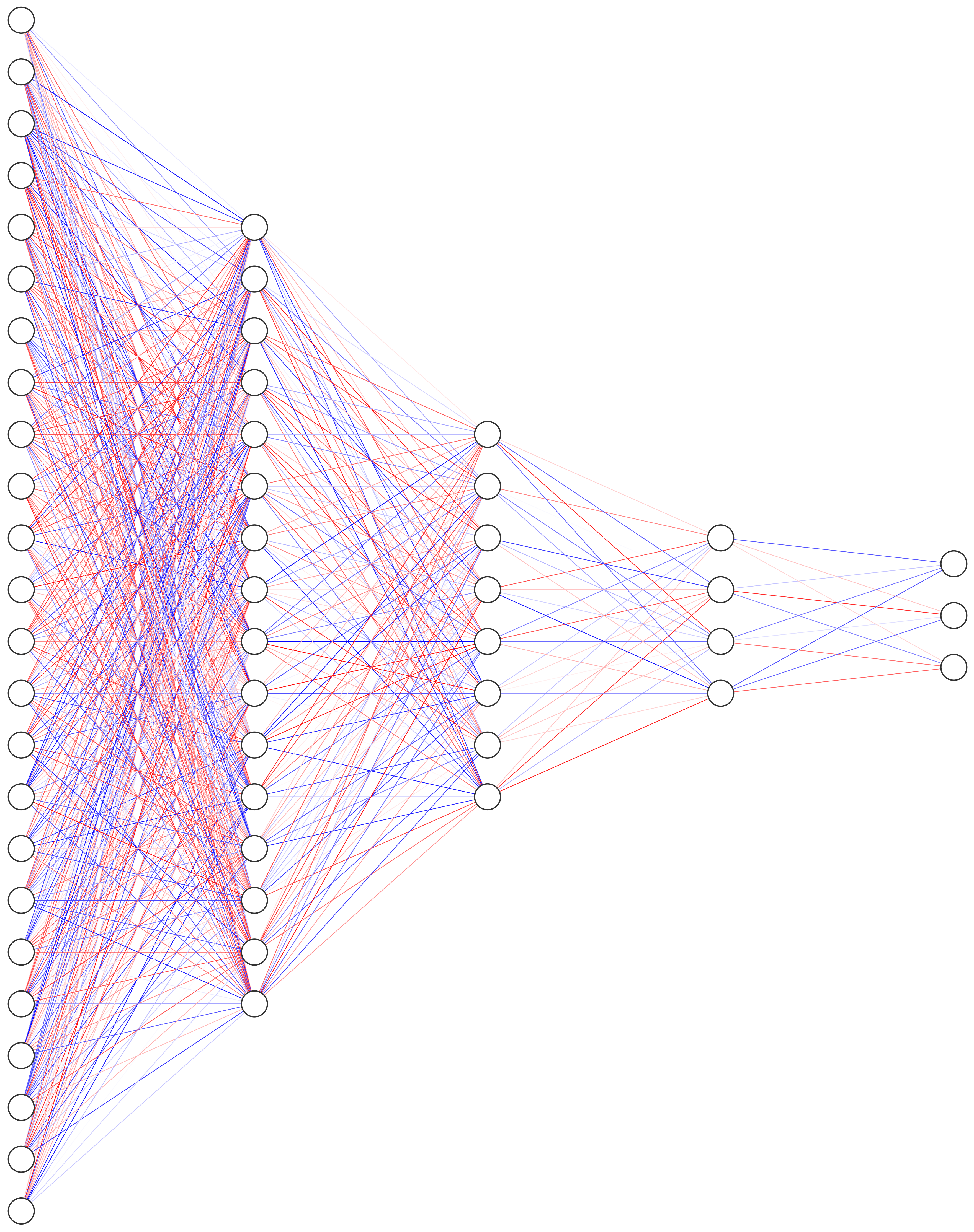


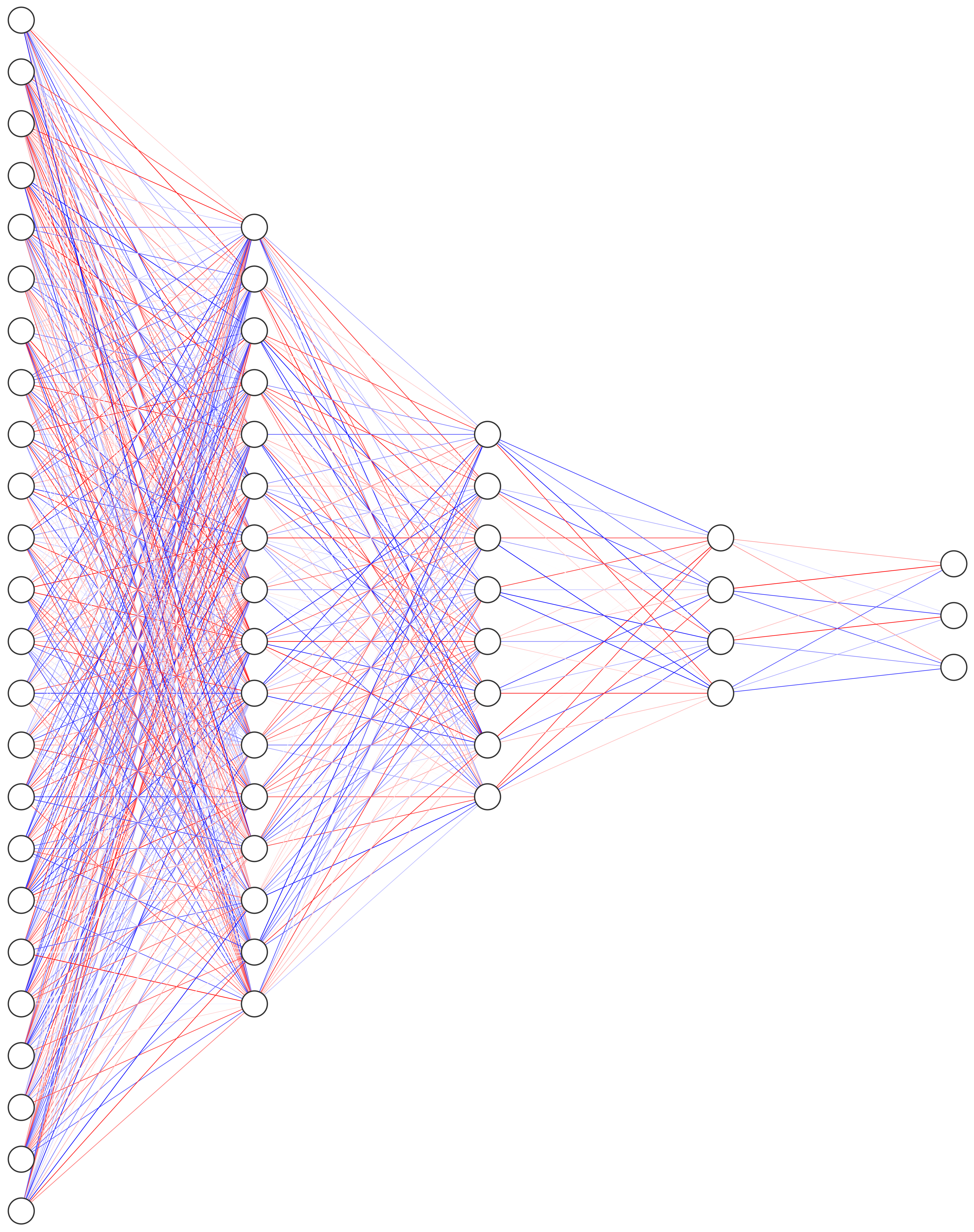


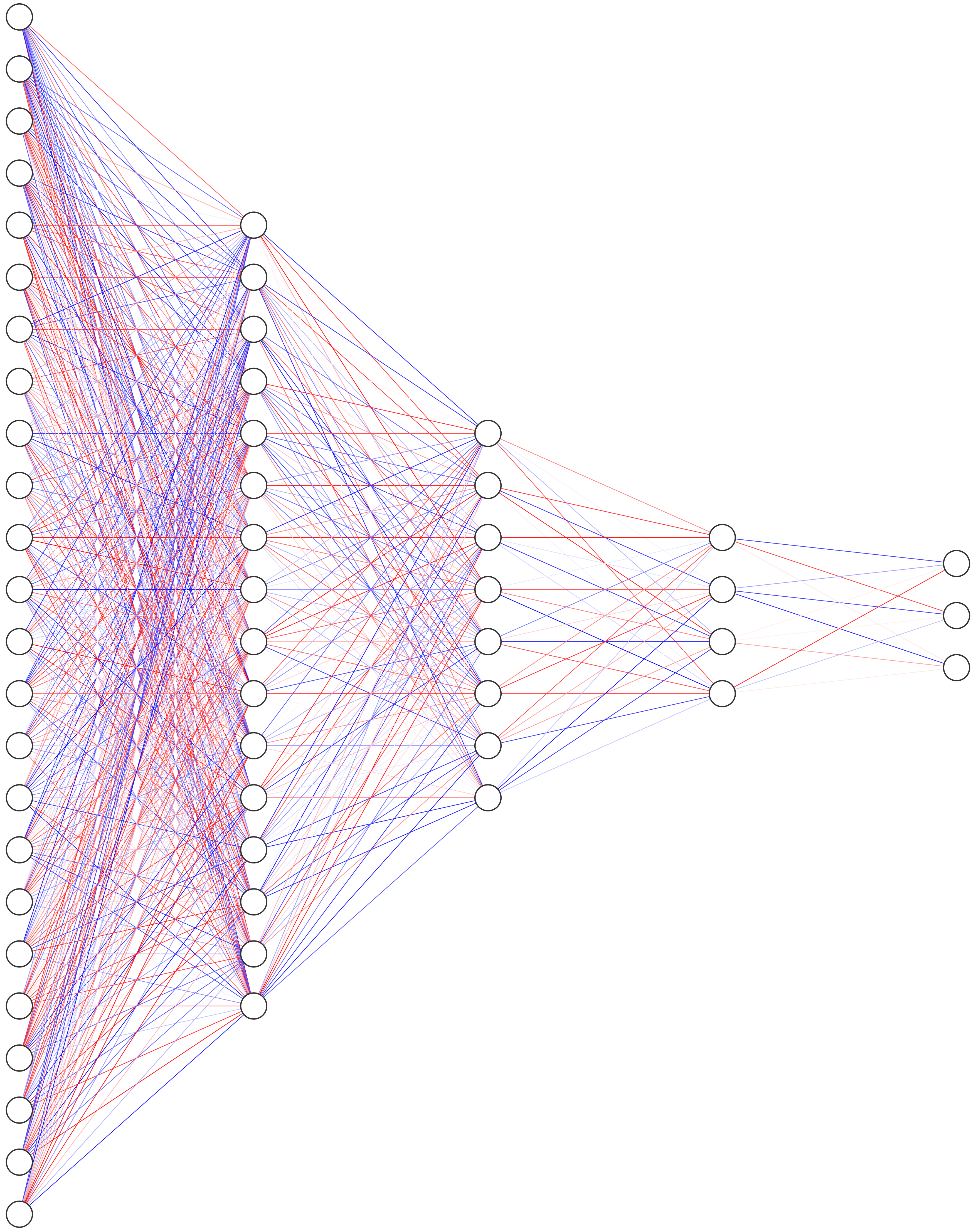


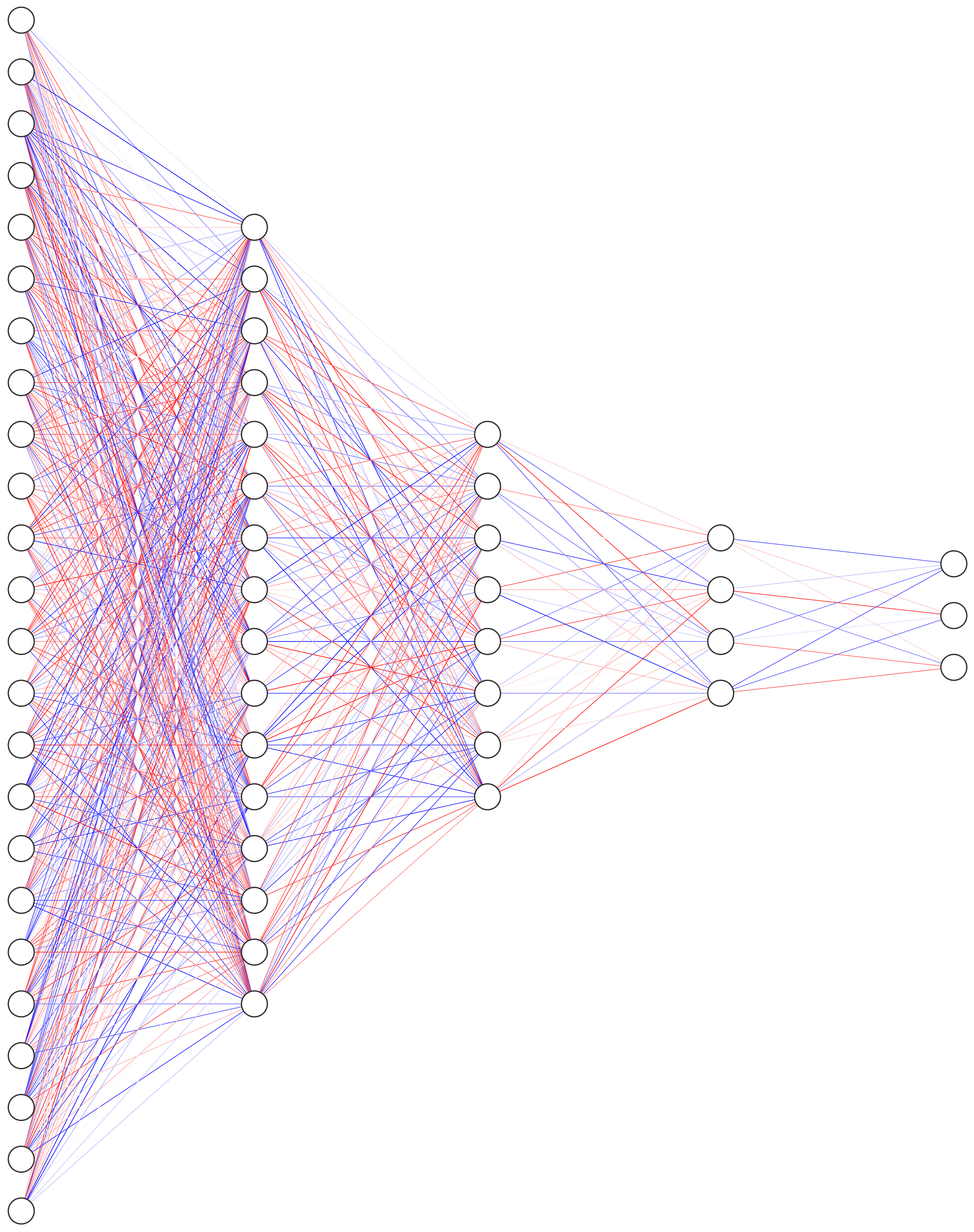


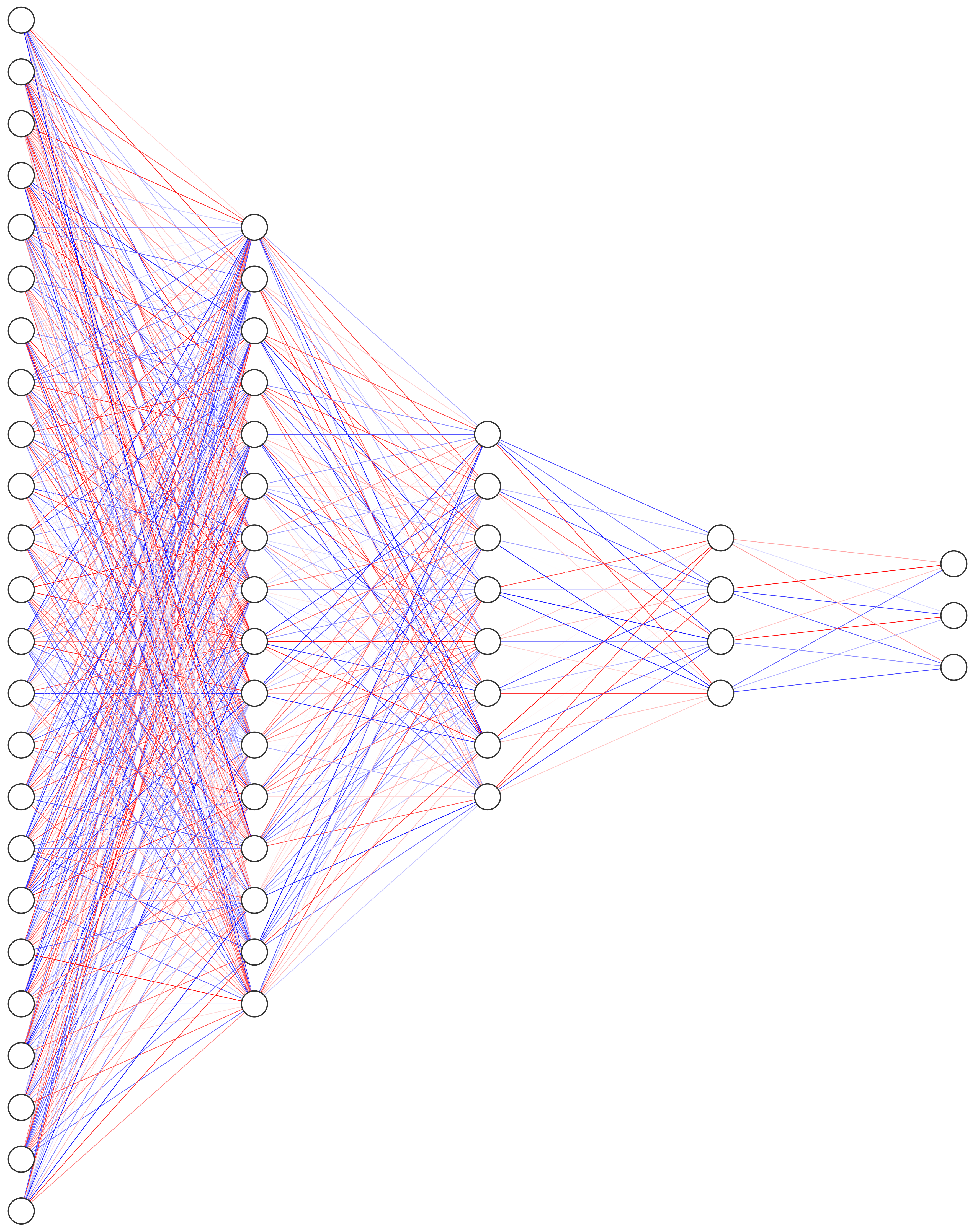




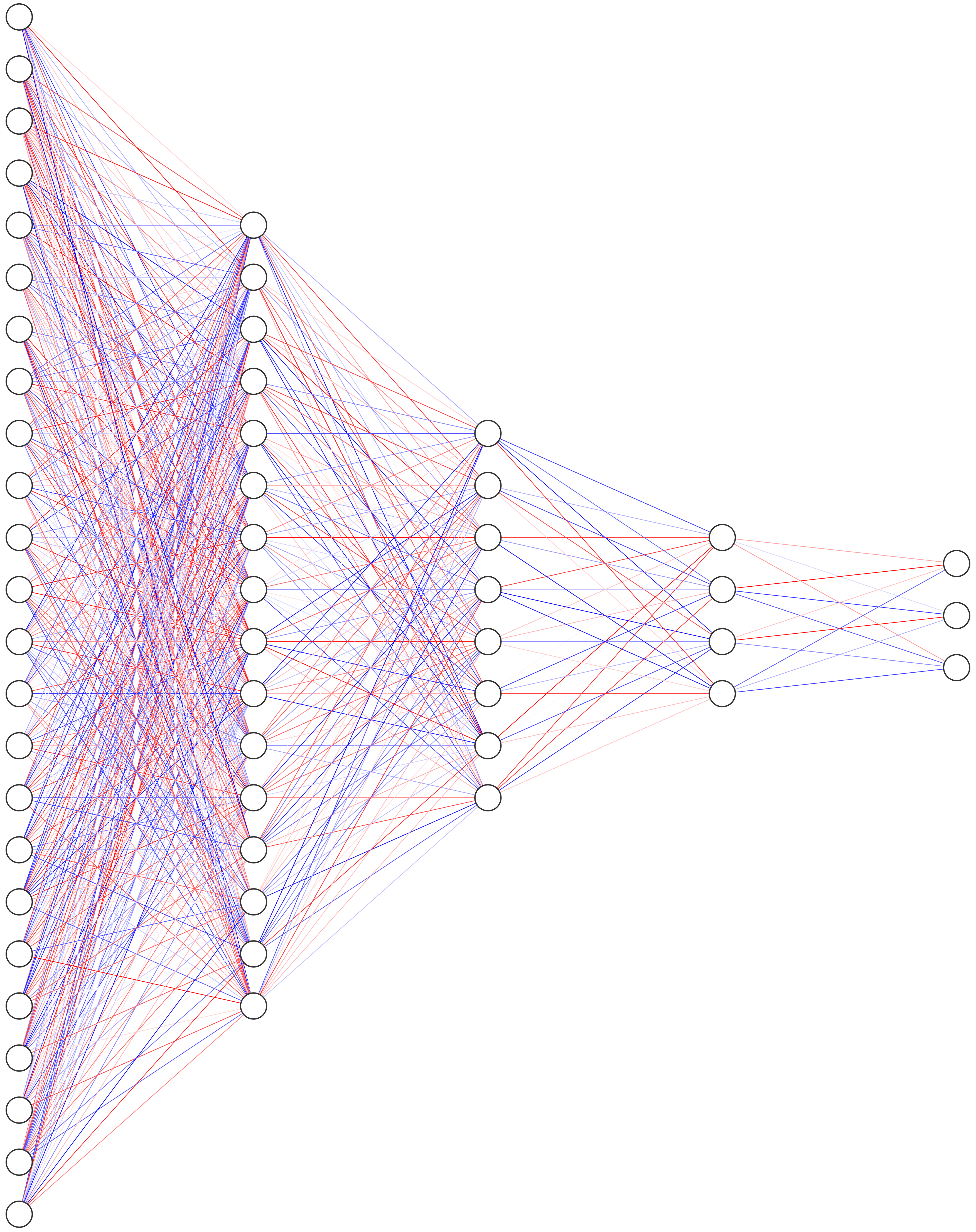




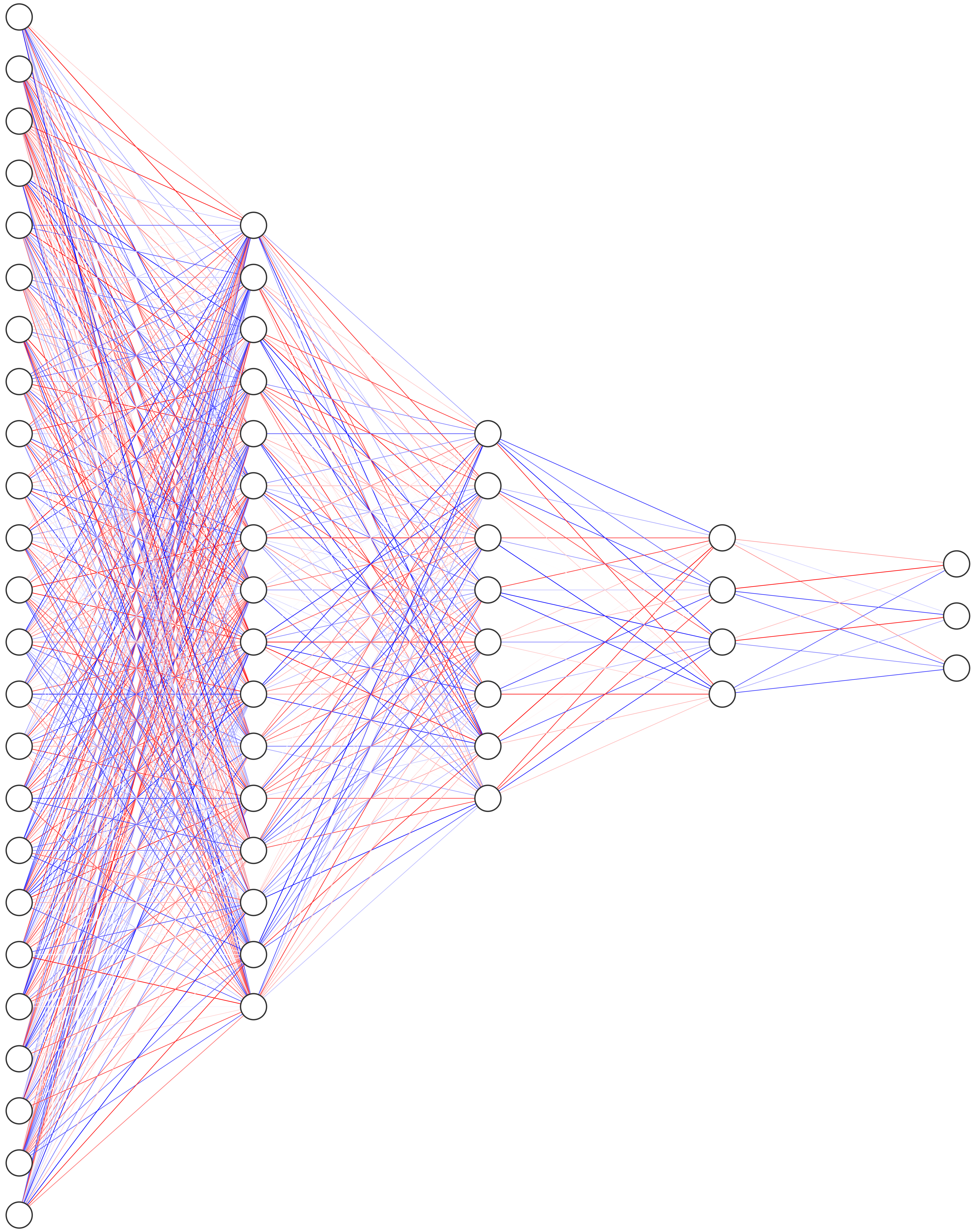




x'

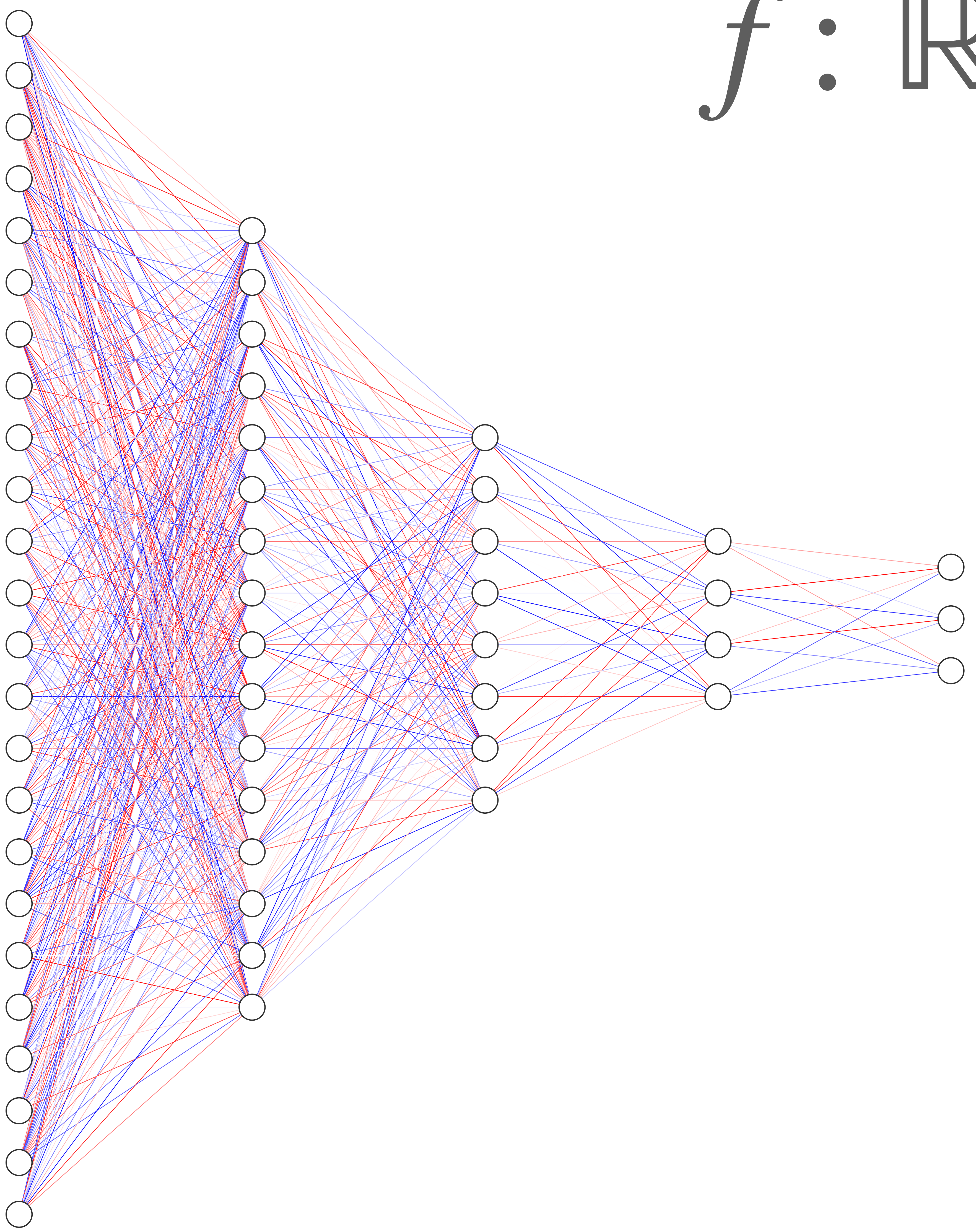


$y = [0.990,\ 0.009,\ 0.001]$



The parameters of a NN are its knowledge.

$$f : \mathbb{R}^n \to \mathbb{R}^m$$



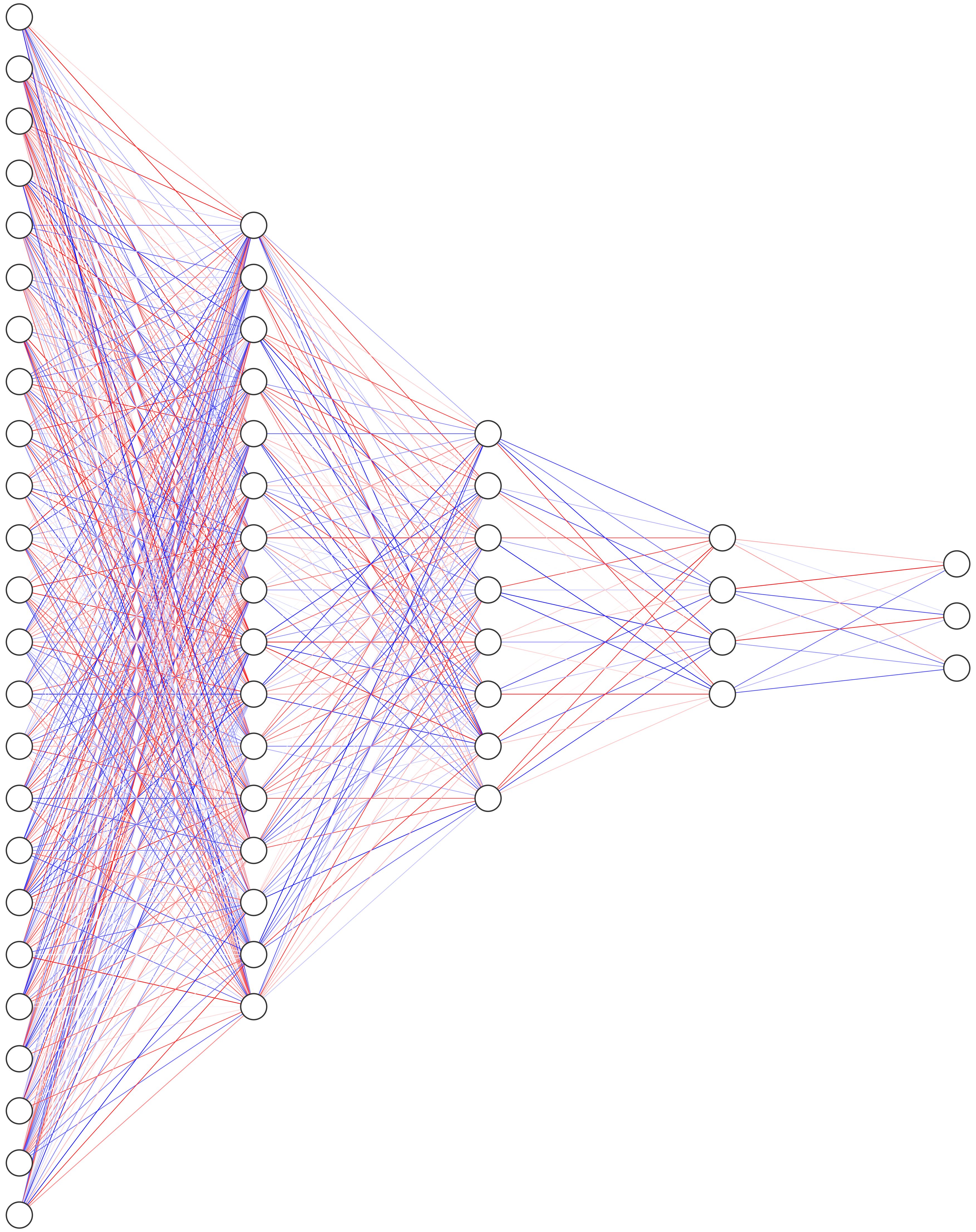
$$f(x) \equiv \text{logits}$$

$$\downarrow$$

$$\phi(f(x)_i) \equiv \frac{\exp(f(x)_i)}{\sum_j \exp(f(x)_j)} \equiv \text{softmax}$$

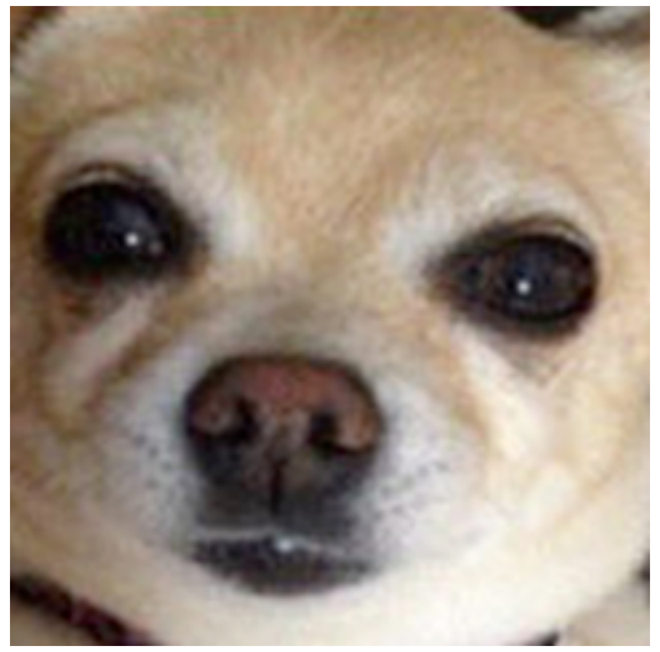
$$\downarrow$$

$$\min_f \mathbb{E}_{x \in X} (\phi(f(x)) - y)^2$$

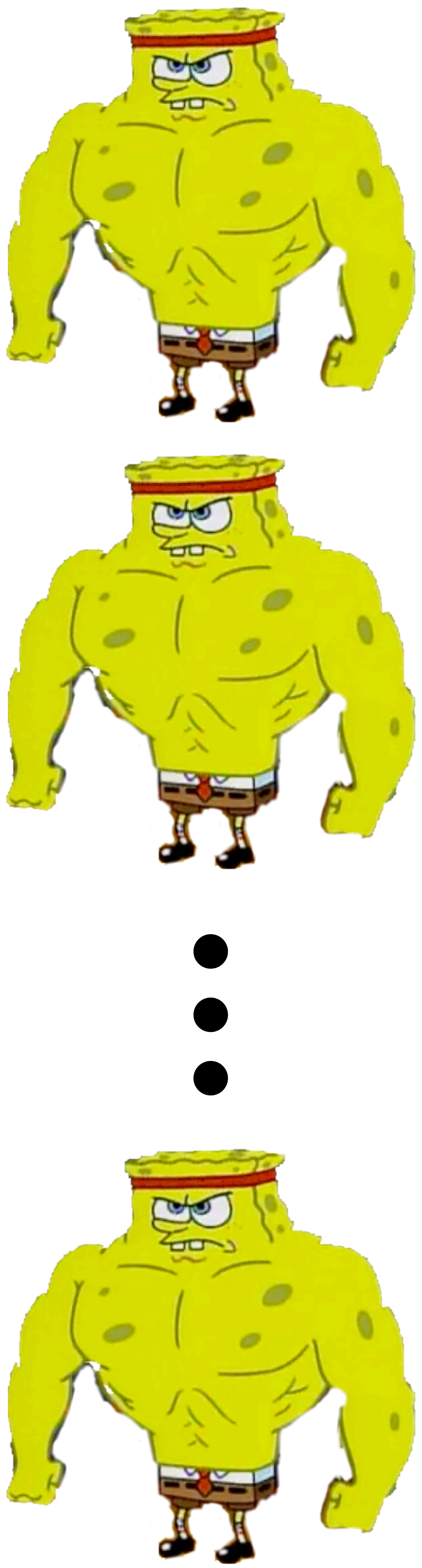


Knowledge

$$\boxed{f(x) \equiv \text{logits}}$$



Trained



$\bar{f}(x)$ - $g(x)$

Trained

⋮

Model Compression

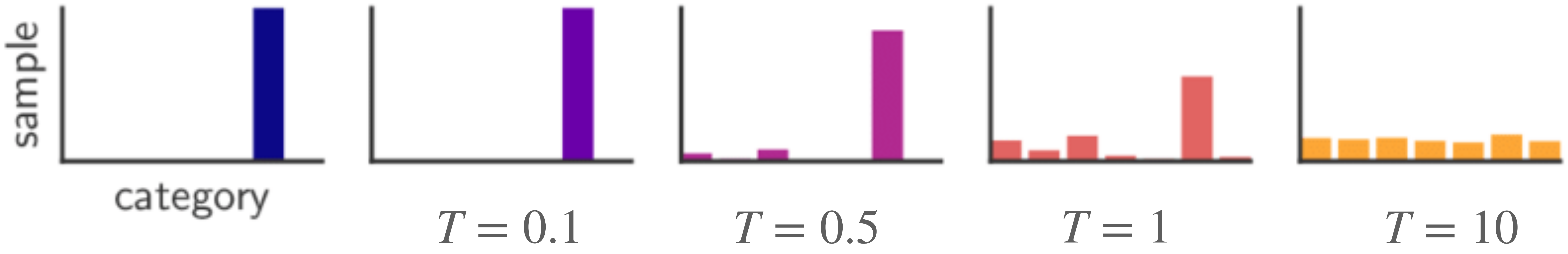
https://doi.org/10.1145/1150402.1150464

$(\bar{f}(x)$ - $g(x))^2$

$$\min_{g} \mathbb{E}(\bar{f}(x) - g(x))^2$$

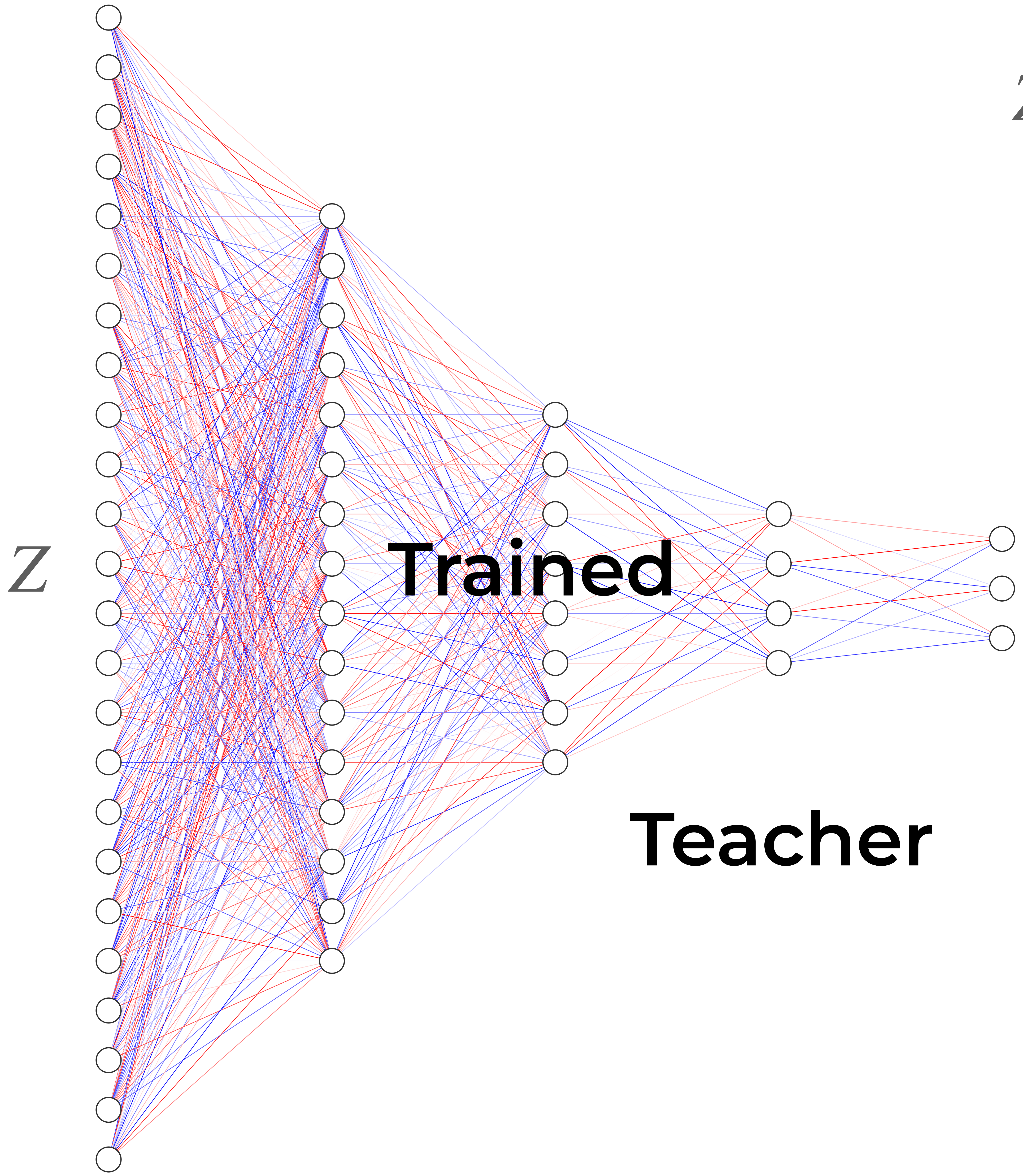
Knowledge Distillation

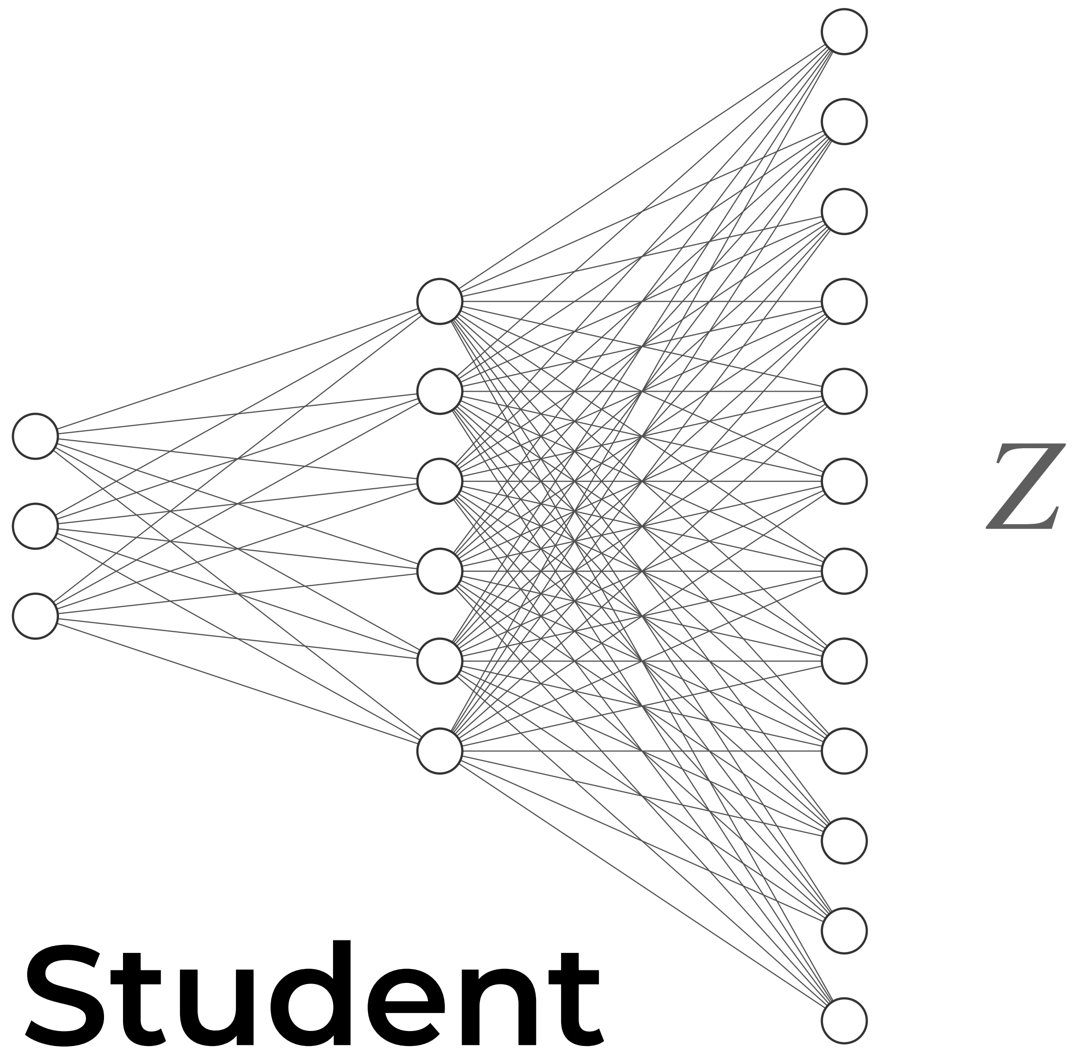
$$\sigma(f(x)_i/T) \equiv \frac{\exp(f(x)_i/T)}{\sum_j \exp(f(x)_j/T)} \xleftrightarrow{Q(\,.\,;\,.\,)} \sigma(g(x)_i/T) \equiv \frac{\exp(g(x)_i/T)}{\sum_j \exp(g(x)_j/T)}$$



http://arxiv.org/abs/1503.02531

$Z =$ Transfer Dataset

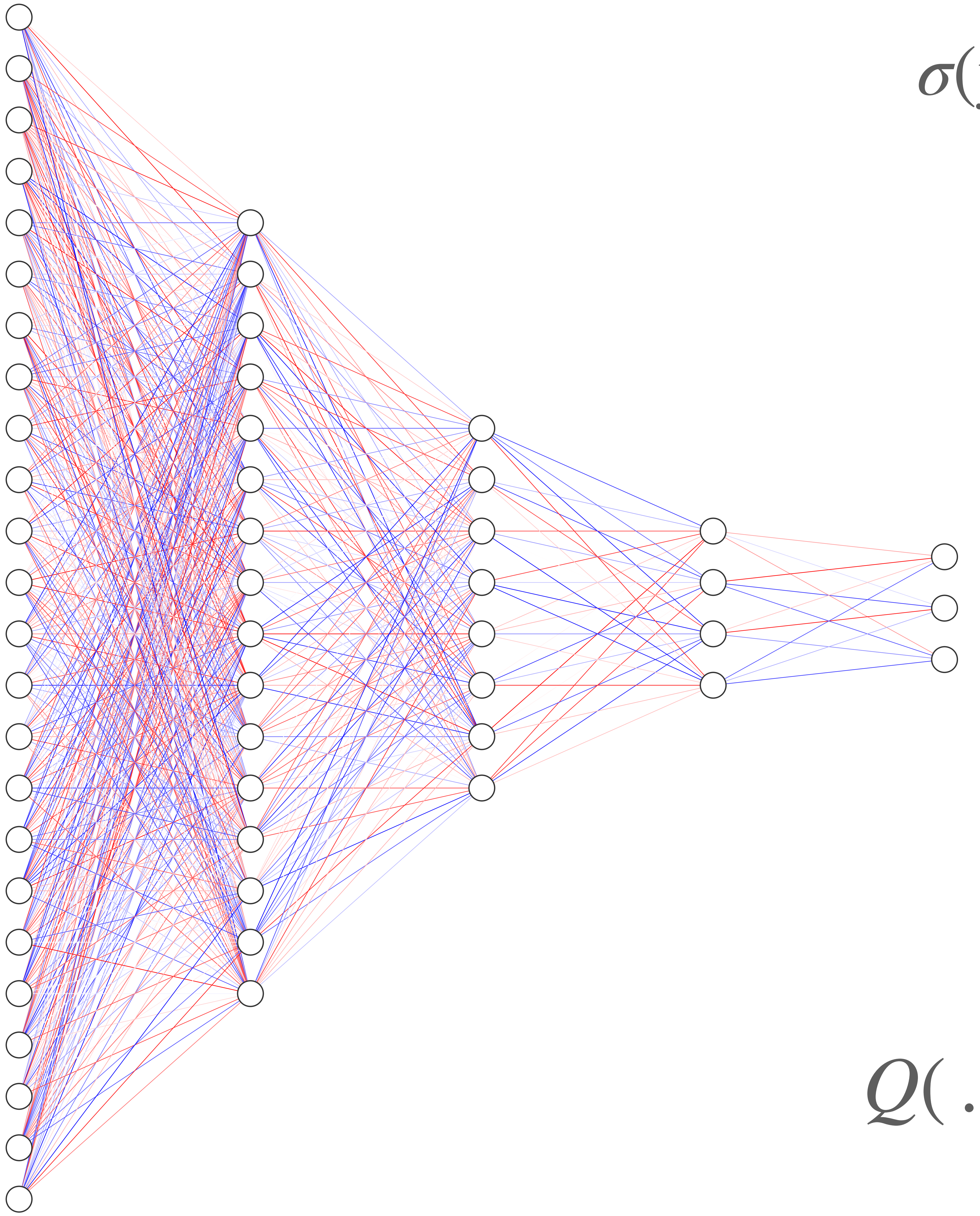




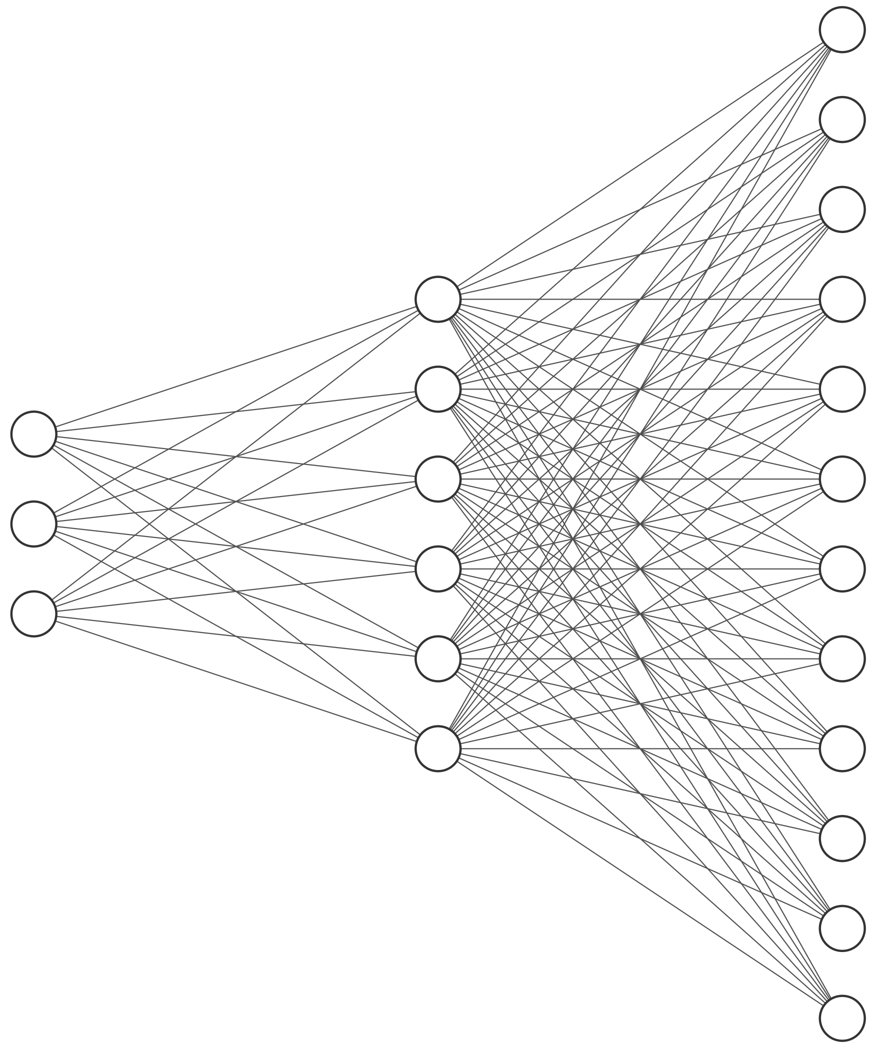
http://arxiv.org/abs/1503.02531

$$\sigma(y_i/T) \equiv \frac{\exp(y_i/T)}{\sum_j \exp(y_j/T)}$$

Z



$$f(z) \longrightarrow \underset{\displaystyle q(z)}{\underset{\equiv}{\sigma(f(z)/T)}} \quad \underset{\displaystyle p(z)}{\underset{\equiv}{\sigma(g(z)/T)}} \longleftarrow g(z)$$

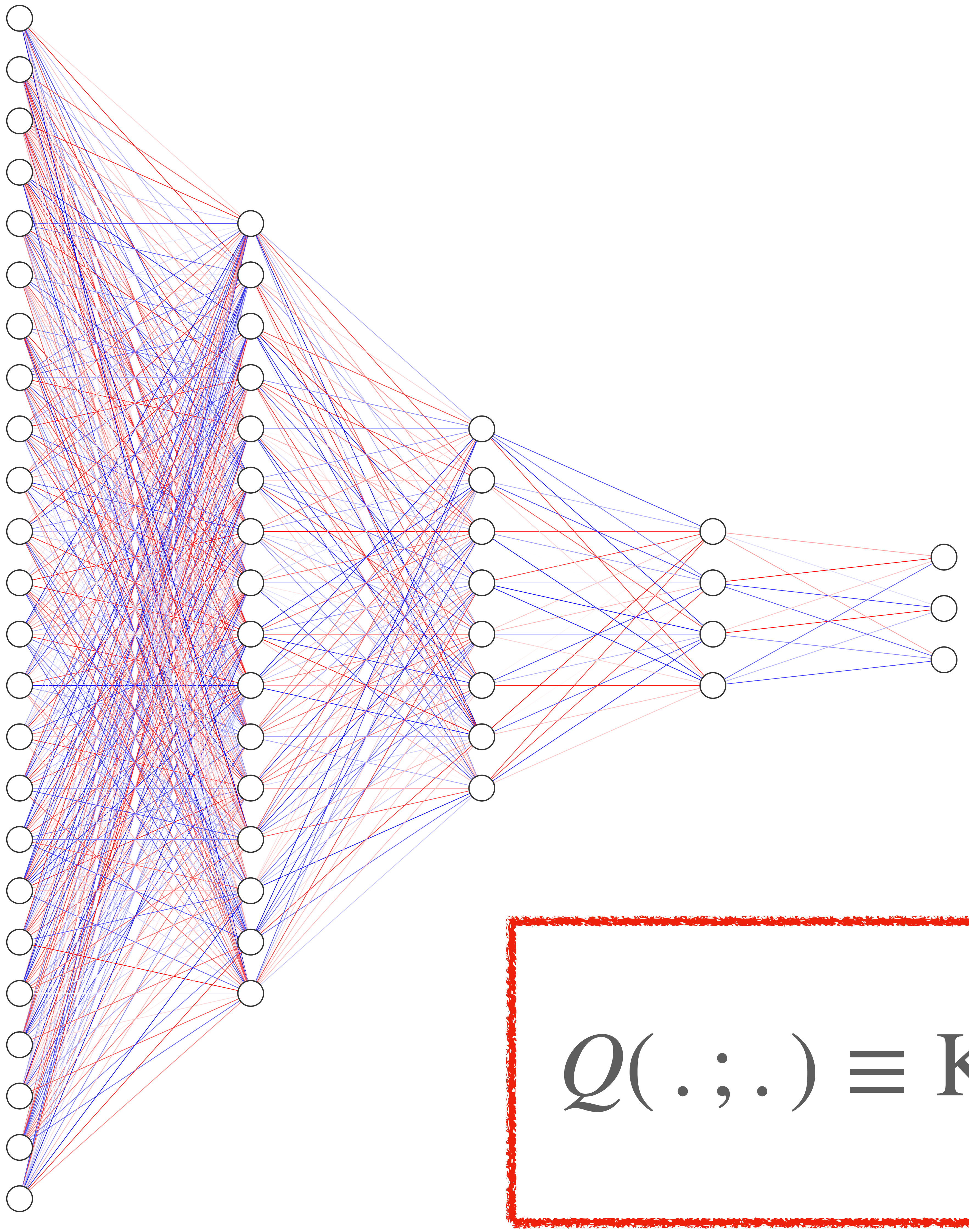


Z

$$Q(\,.\,;\,.\,) \equiv \mathrm{H}_q = -\,\mathbb{E}_{x\sim p(\cdot)}[\log q(x)]$$

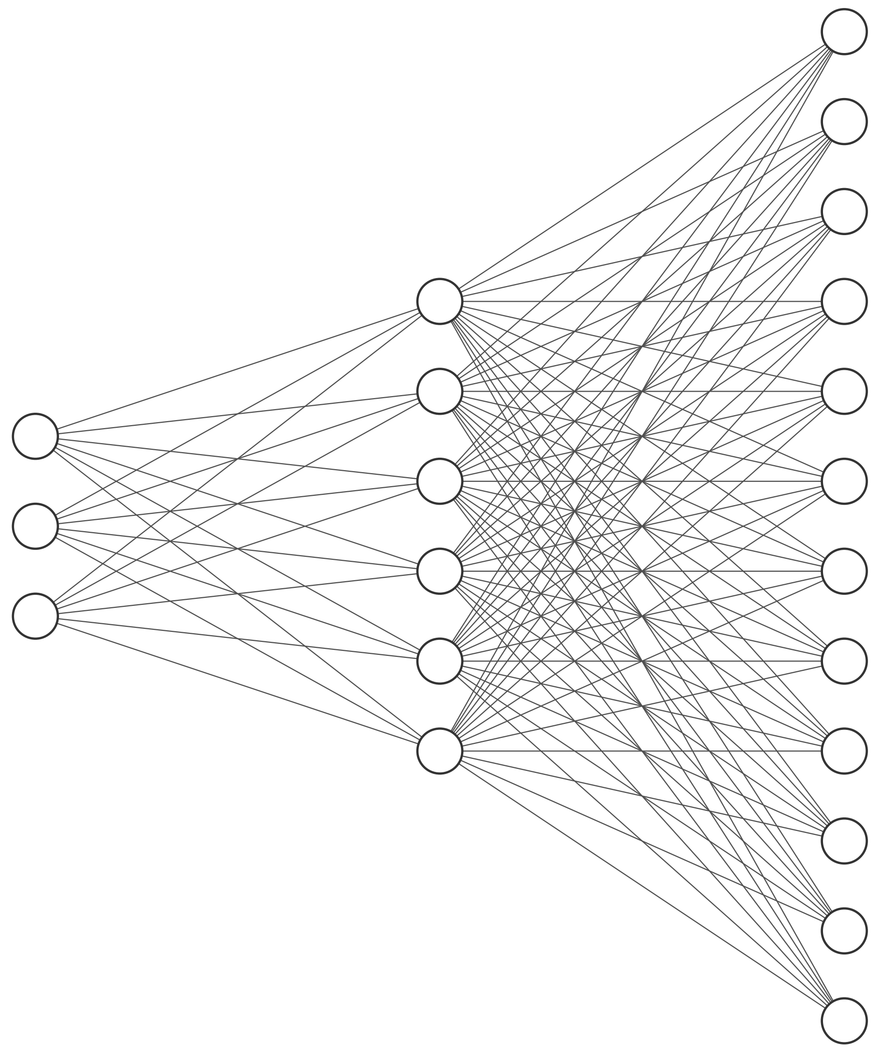
http://arxiv.org/abs/1503.02531

Z



$$f(z) \longrightarrow \sigma(f(z)/T) \qquad \sigma(g(z)/T) \longleftarrow g(z)$$

$$\sigma(f(z)/T) \equiv q(z) \qquad \sigma(g(z)/T) \equiv p(z)$$



Z

$$\boxed{Q(\,.\,;\,.\,) \equiv \mathrm{KL}(p\,||\,q) = -\,\mathbb{E}_{z\sim p(\cdot)}\left[\log \frac{q(z)}{p(z)}\right]}$$

http://arxiv.org/abs/1503.02531

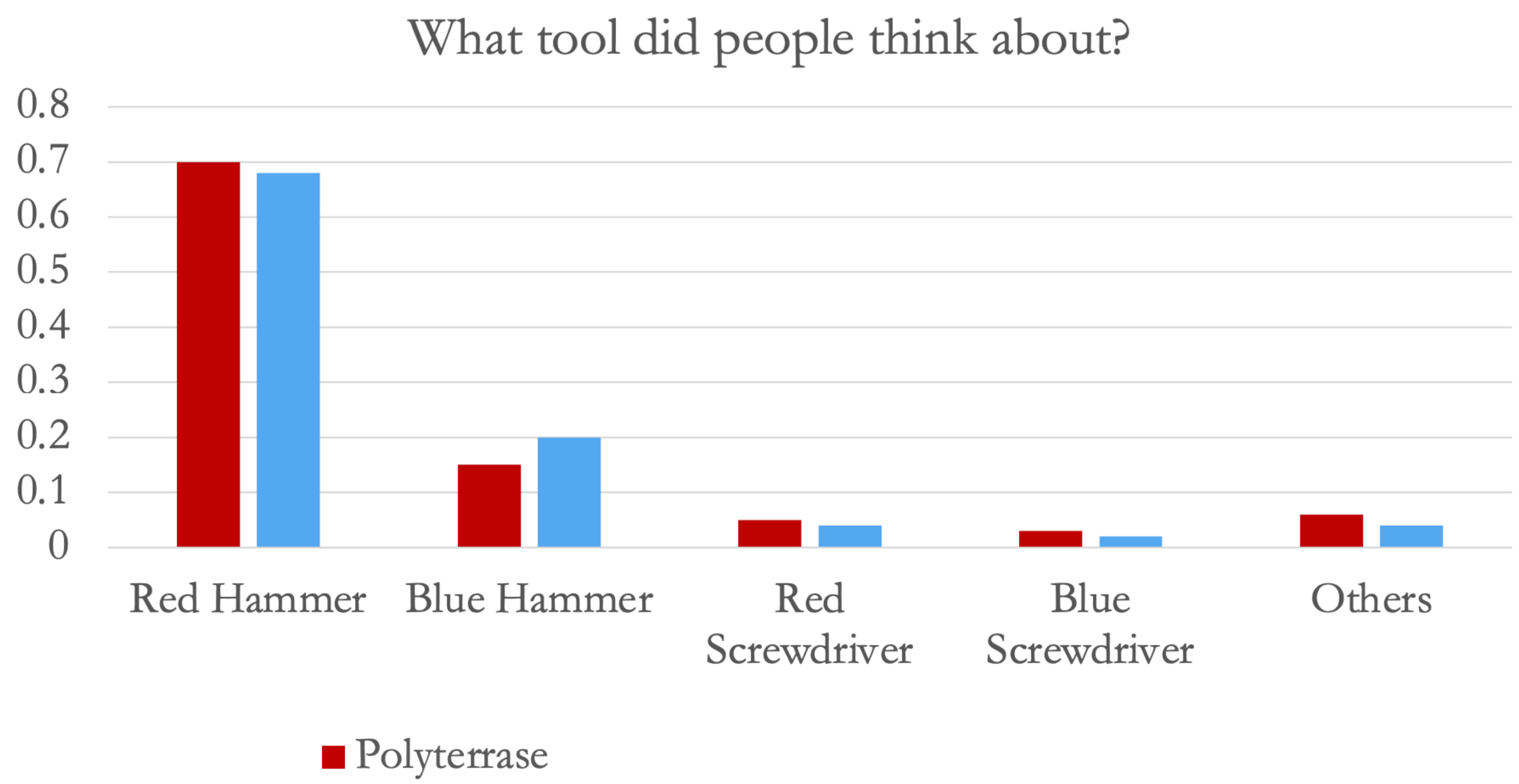
Quick! Answer the following questions ASAP!

- ✓ 32 + 6
- ✓ 2 * 125
- ✓ 12 * 12
- ✓ 25 * 51
- ✓ 100*11 + 22

Quick! Answer the following questions ASAP!

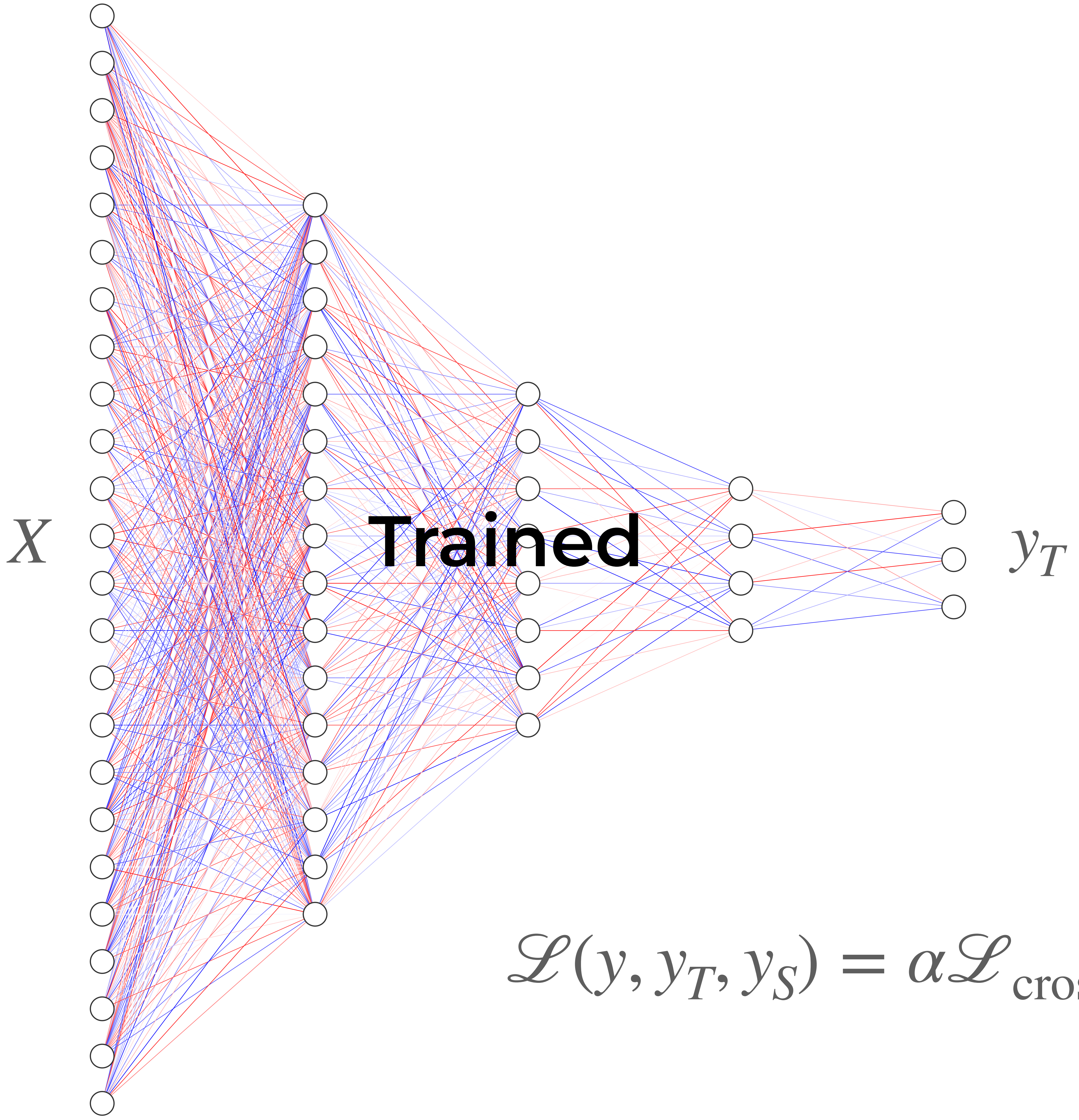
- ✓ 32 + 6
- ✓ 2 * 125
- ✓ 12 * 12
- ✓ 25 * 51
- ✓ 100*11 + 22

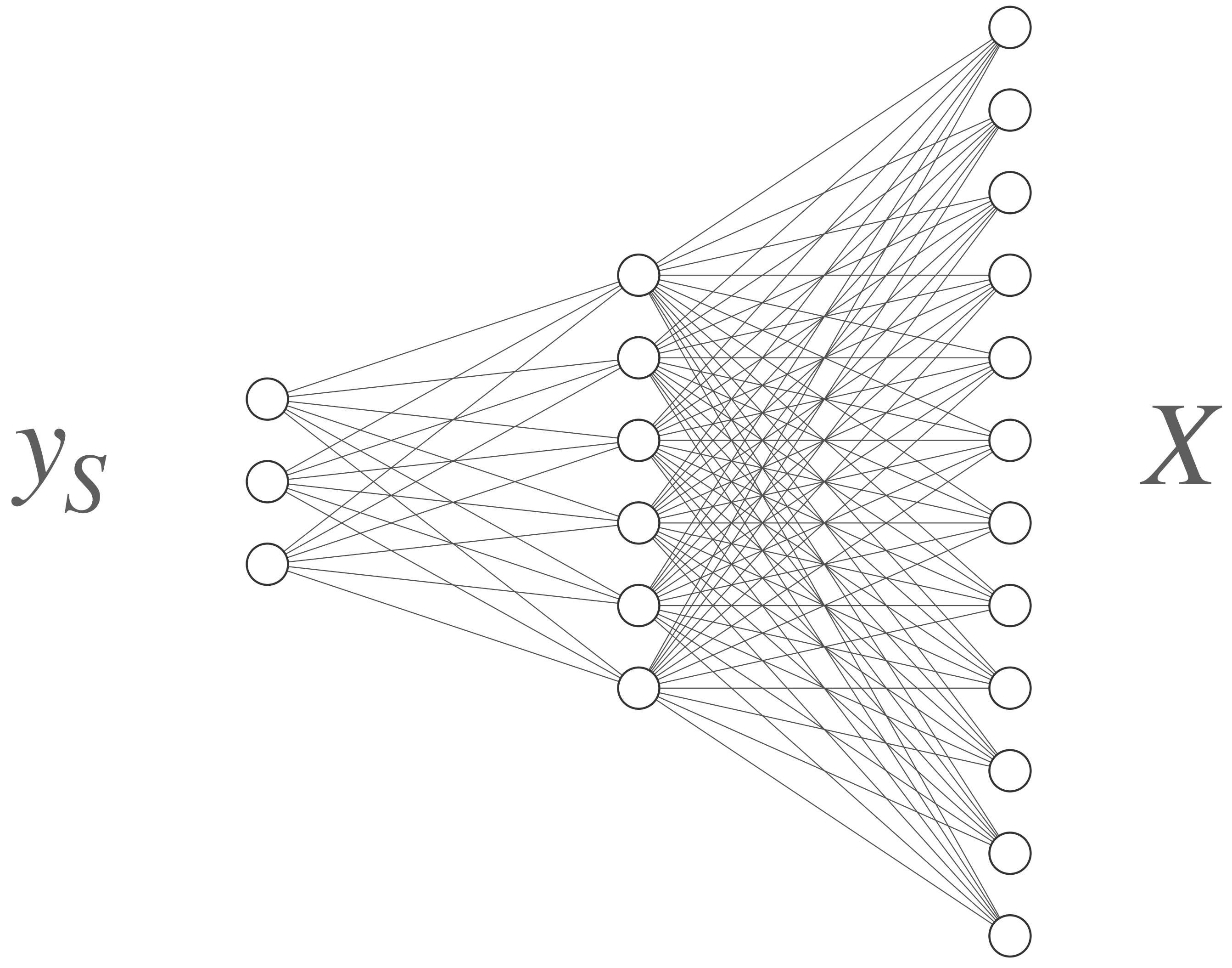
Done? Think about a COLOUR and a TOOL!



$$\mathrm{KL}(p\,||\,q) = -\,\mathbb{E}_{x\sim p(\cdot)}\left[\log \frac{q(x)}{p(x)}\right]$$

Credit: Jesus Solano



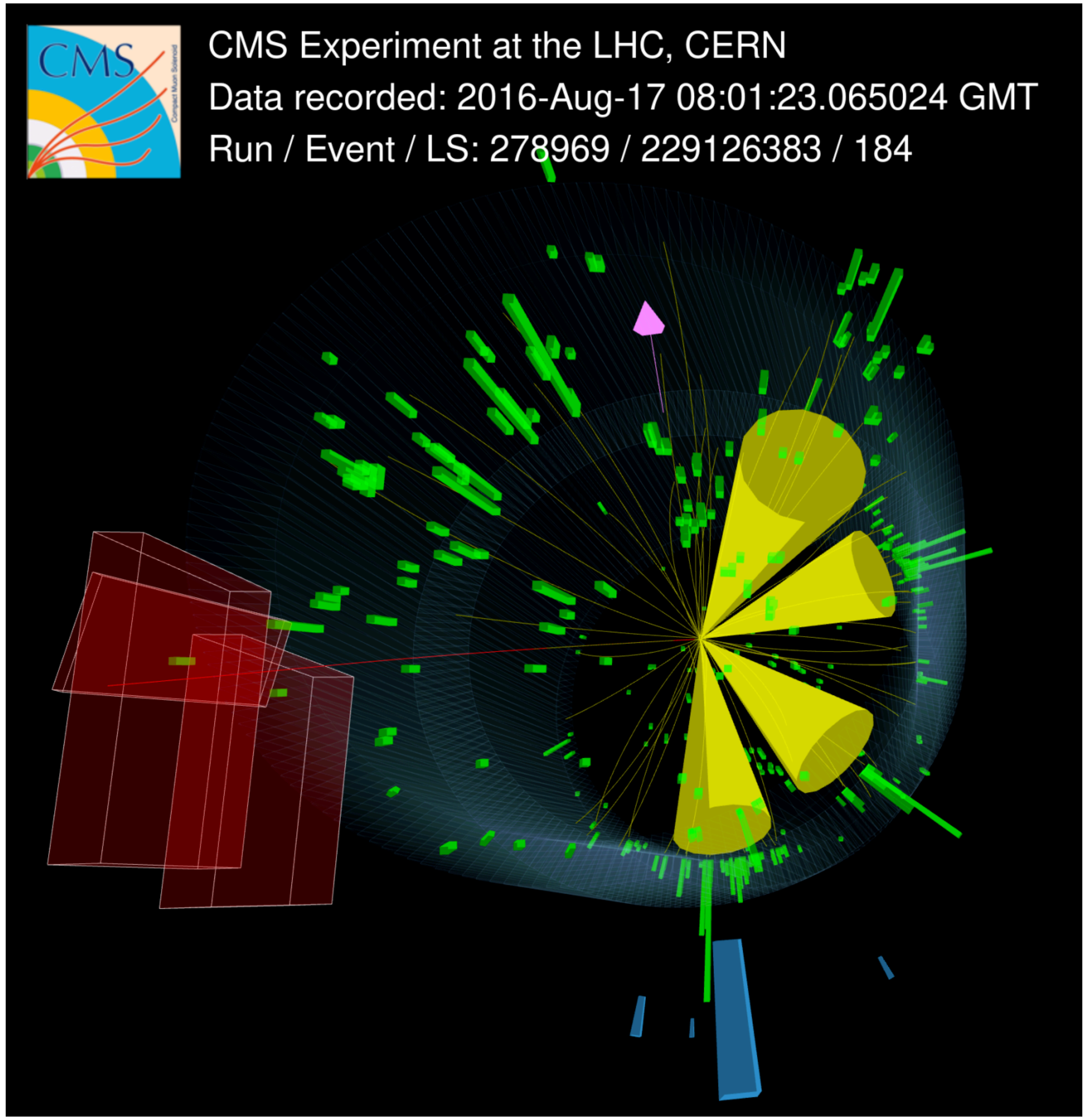


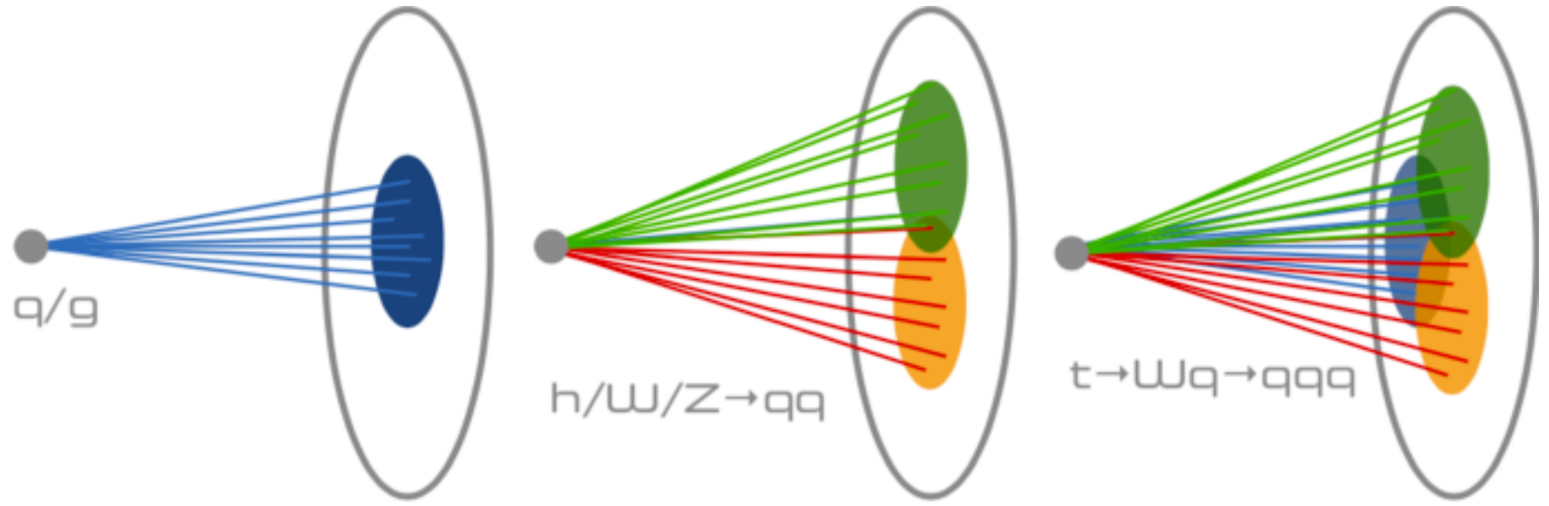
$$\mathcal{L}(y, y_T, y_S) = \alpha \mathcal{L}_{\text{cross-entropy}}(y, y_S) + (1 - \alpha)\mathcal{L}_{\text{distillaion}}(y_T, y_S)$$

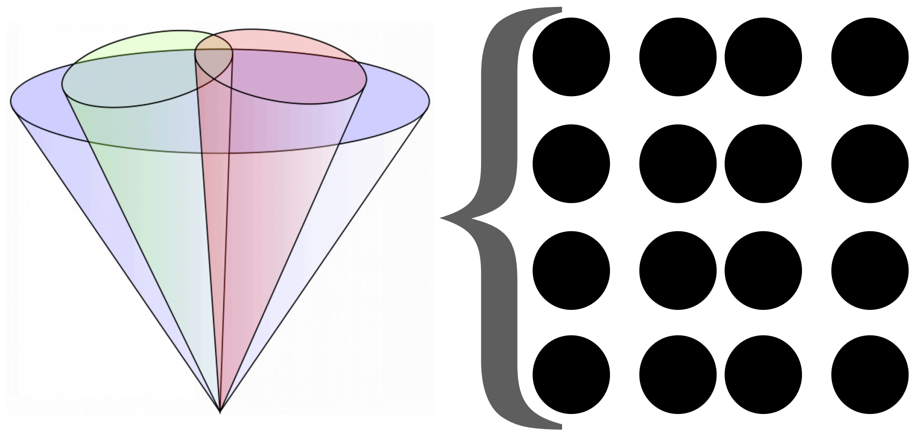
$$\mathcal{L}_{\text{distillaion}}(y_T, y_S) = \mathrm{D}_{\mathrm{KL}}(\sigma(y_S/T)\,||\,\sigma(y_T/T)) \times T^2$$

http://arxiv.org/abs/1503.02531
https://arxiv.org/abs/1805.04770

My Knowledge Distillation Adventure





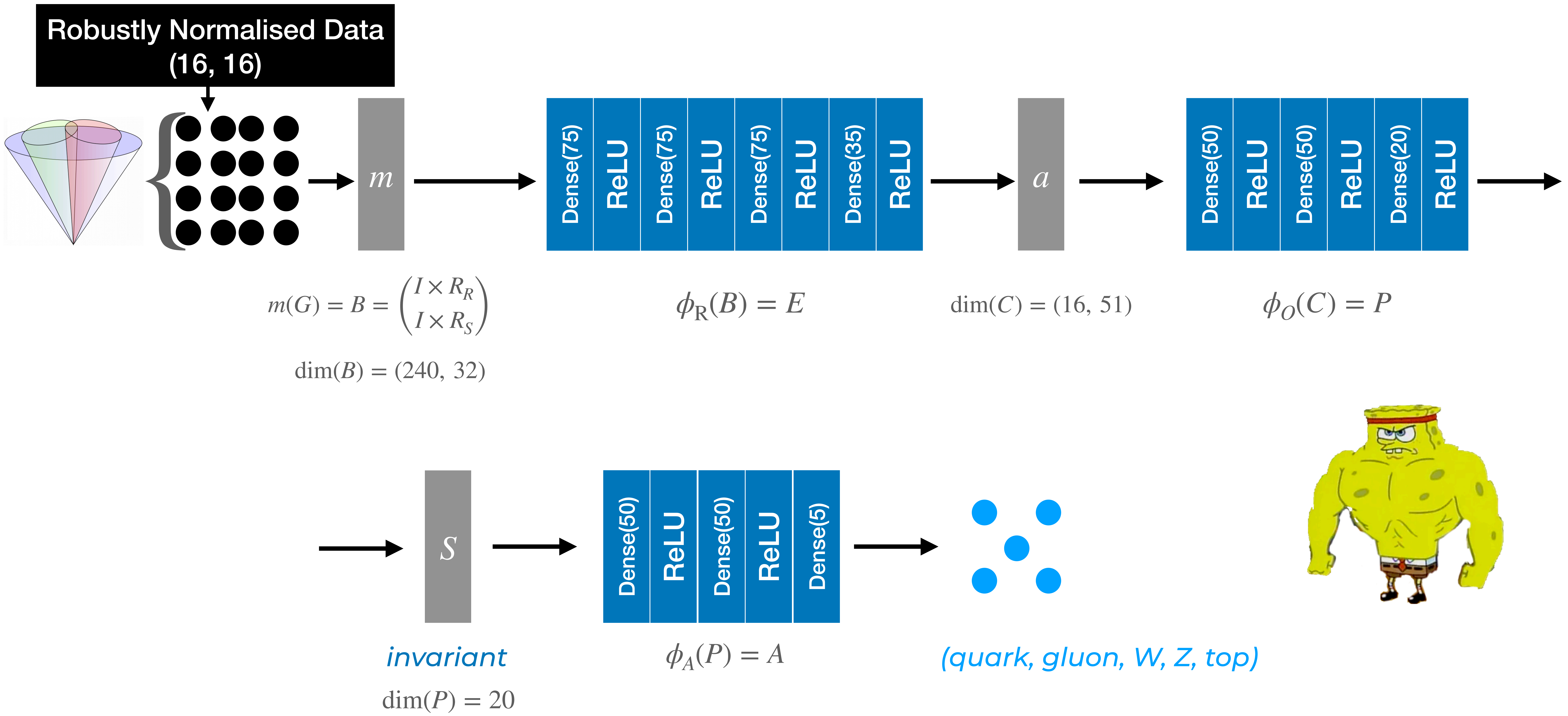


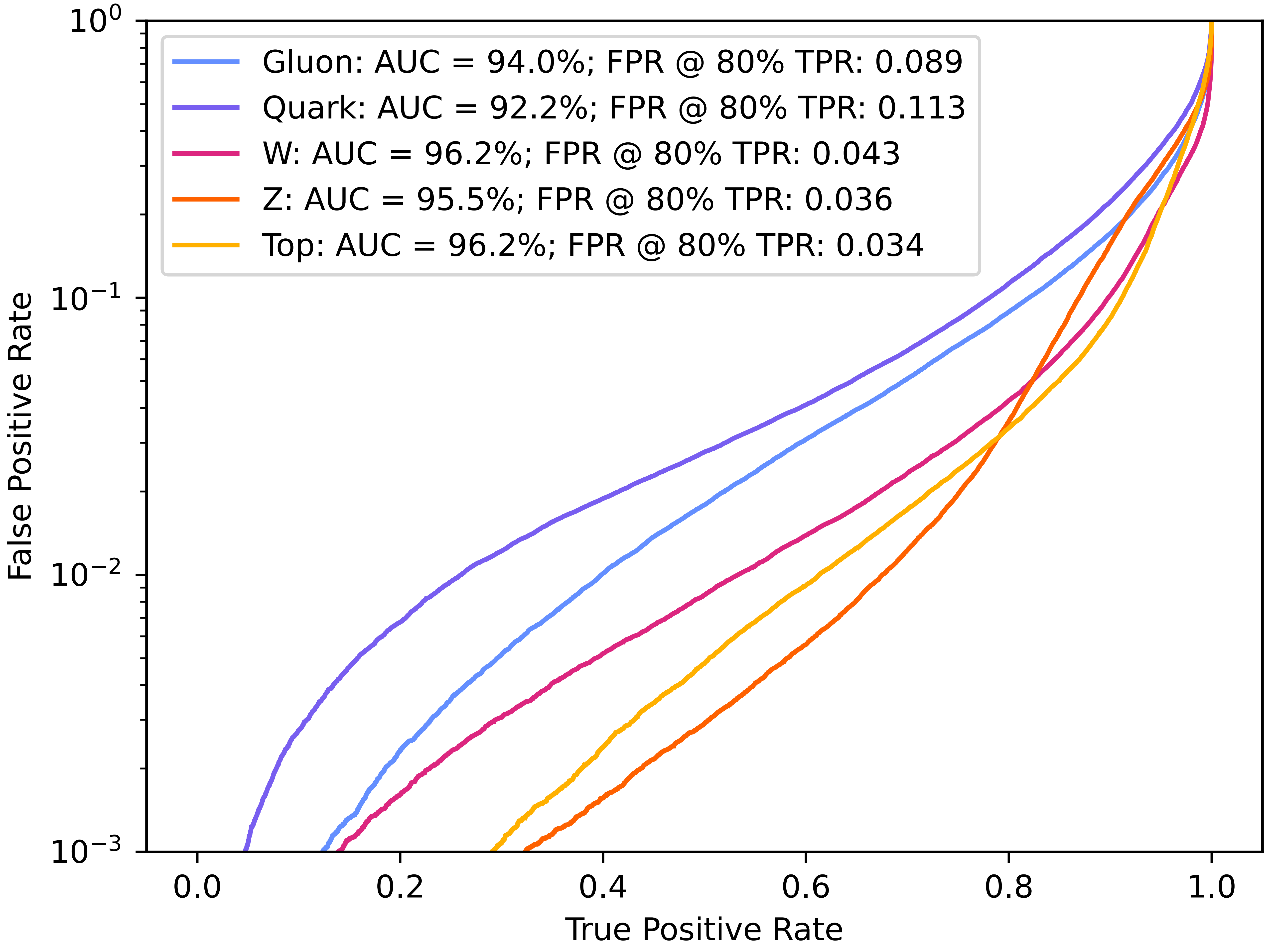
quarks
gluons
Ws
Zs
tops

Training: 572 000 jets | 20% validation
Testing: 282 000 jets

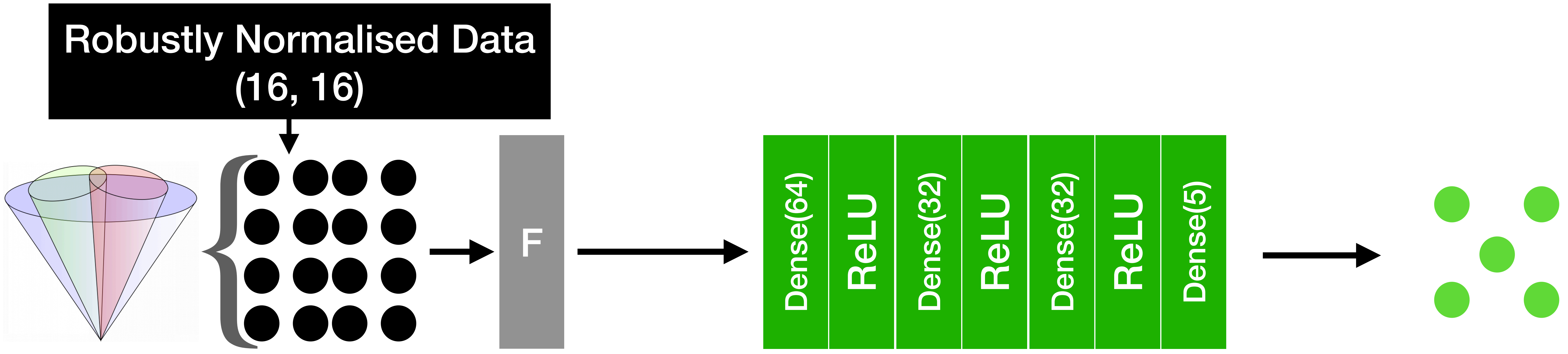
equally distributed between 5 classes:
quarks, gluons, Ws, Zs, tops

Every jet defined by maximum **150** constituents.
Every constituent has a maximum of **16** features.


Robustly Normalised Data
(16, 16)
m
Dense(75)
ReLU
Dense(75)
ReLU
Dense(75)
ReLU
Dense(35)
ReLU
a
Dense(50)
ReLU
Dense(50)
ReLU
Dense(20)
ReLU
$m(G) = B = \begin{pmatrix} I \times R_R \\ I \times R_S \end{pmatrix}$
$\dim(B) = (240,\ 32)$
$\phi_{\mathrm{R}}(B) = E$
$\dim(C) = (16,\ 51)$
$\phi_O(C) = P$
S
Dense(50)
ReLU
Dense(50)
ReLU
Dense(5)
invariant
$\dim(P) = 20$
$\phi_A(P) = A$
(quark, gluon, W, Z, top)


Gluon: AUC = 94.0%; FPR @ 80% TPR: 0.089
Quark: AUC = 92.2%; FPR @ 80% TPR: 0.113
W: AUC = 96.2%; FPR @ 80% TPR: 0.043
Z: AUC = 95.5%; FPR @ 80% TPR: 0.036
Top: AUC = 96.2%; FPR @ 80% TPR: 0.034
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0

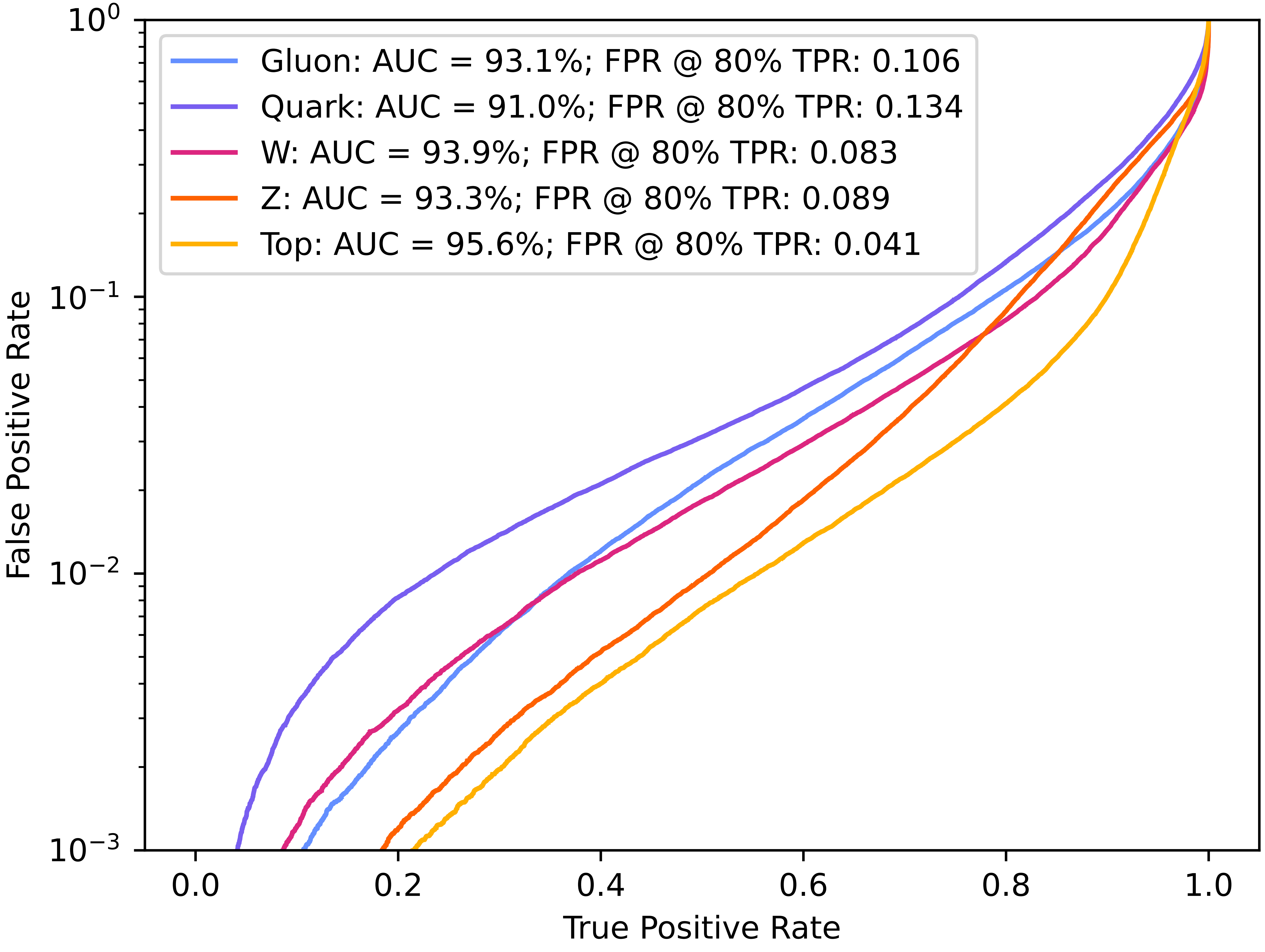
Appropriately Small Student



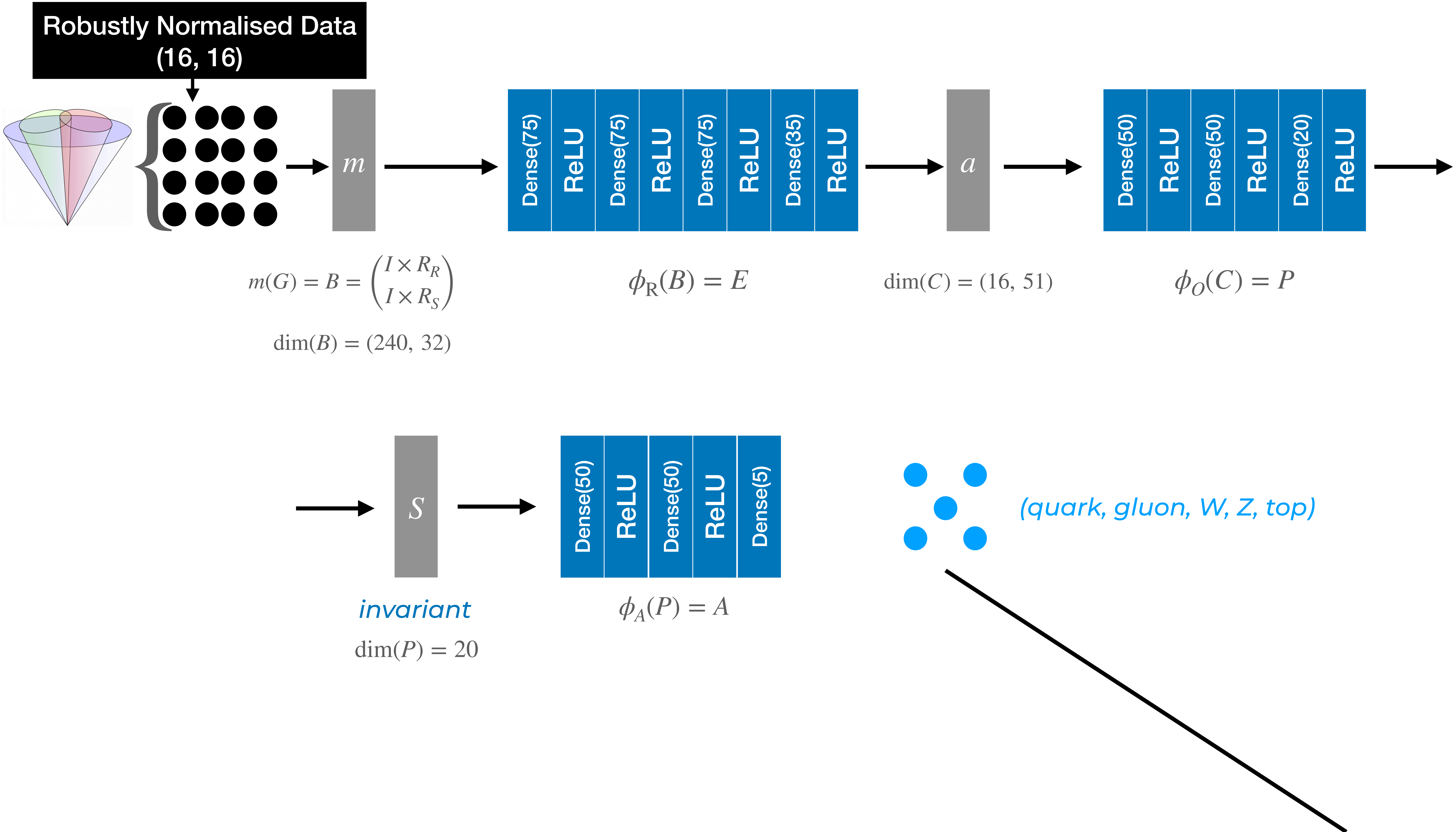
$\dim(I) = 256$

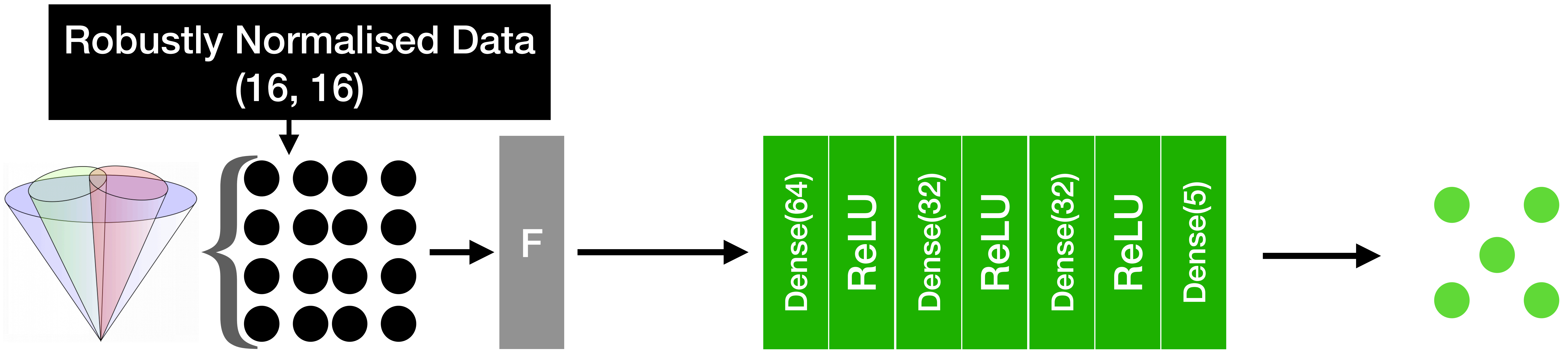
$\phi_S(I) = A$

(quark, gluon, W, Z, top)



Gluon: AUC = 93.1%; FPR @ 80% TPR: 0.106
Quark: AUC = 91.0%; FPR @ 80% TPR: 0.134
W: AUC = 93.9%; FPR @ 80% TPR: 0.083
Z: AUC = 93.3%; FPR @ 80% TPR: 0.089
Top: AUC = 95.6%; FPR @ 80% TPR: 0.041
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0

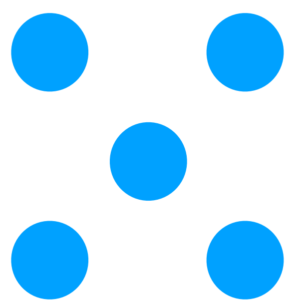

Robustly Normalised Data
(16, 16)
m
Dense(75)
ReLU
Dense(75)
ReLU
Dense(75)
ReLU
Dense(35)
ReLU
a
Dense(50)
ReLU
Dense(50)
ReLU
Dense(20)
ReLU
$m(G) = B = \begin{pmatrix} I \times R_R \\ I \times R_S \end{pmatrix}$
$\dim(B) = (240,\ 32)$
$\phi_{\mathrm{R}}(B) = E$
$\dim(C) = (16,\ 51)$
$\phi_O(C) = P$
S
Dense(50)
ReLU
Dense(50)
ReLU
Dense(5)
(quark, gluon, W, Z, top)
invariant
$\dim(P) = 20$
$\phi_A(P) = A$

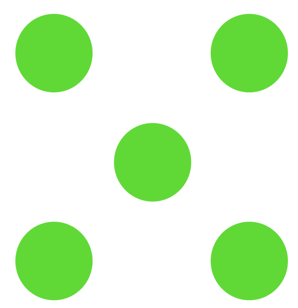


$\dim(I) = 256$

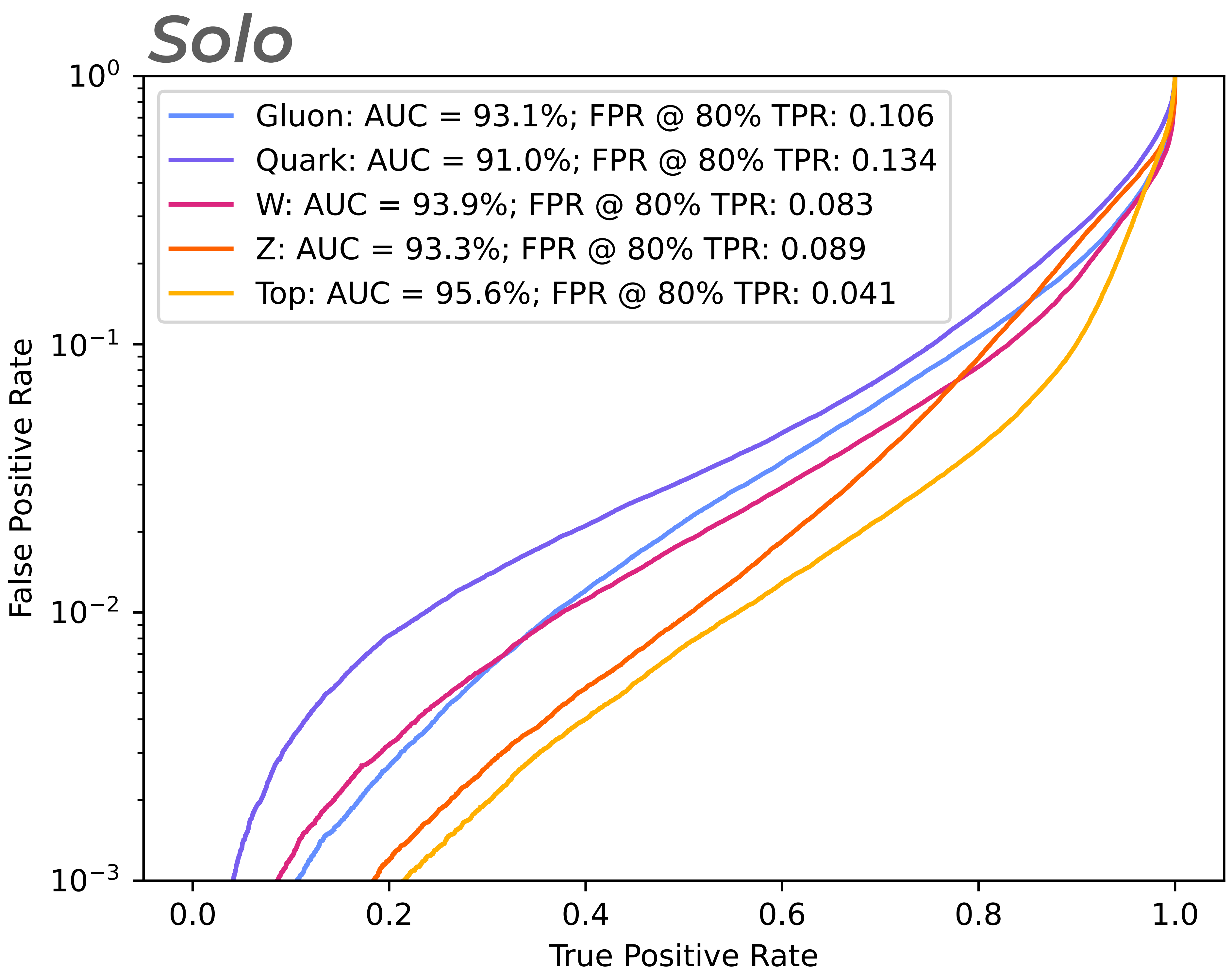
$\phi_S(I) = A$

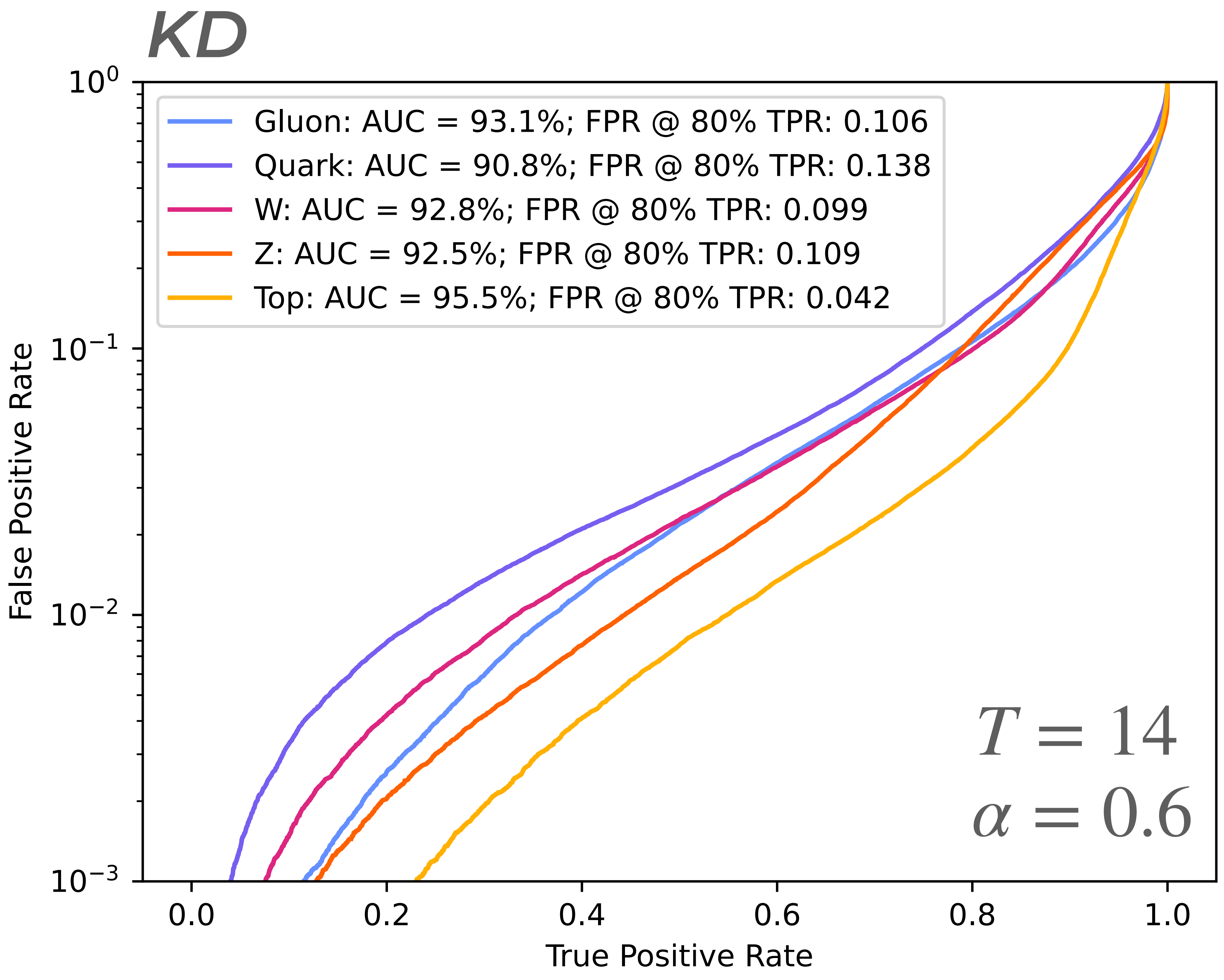
(quark, gluon, W, Z, top)

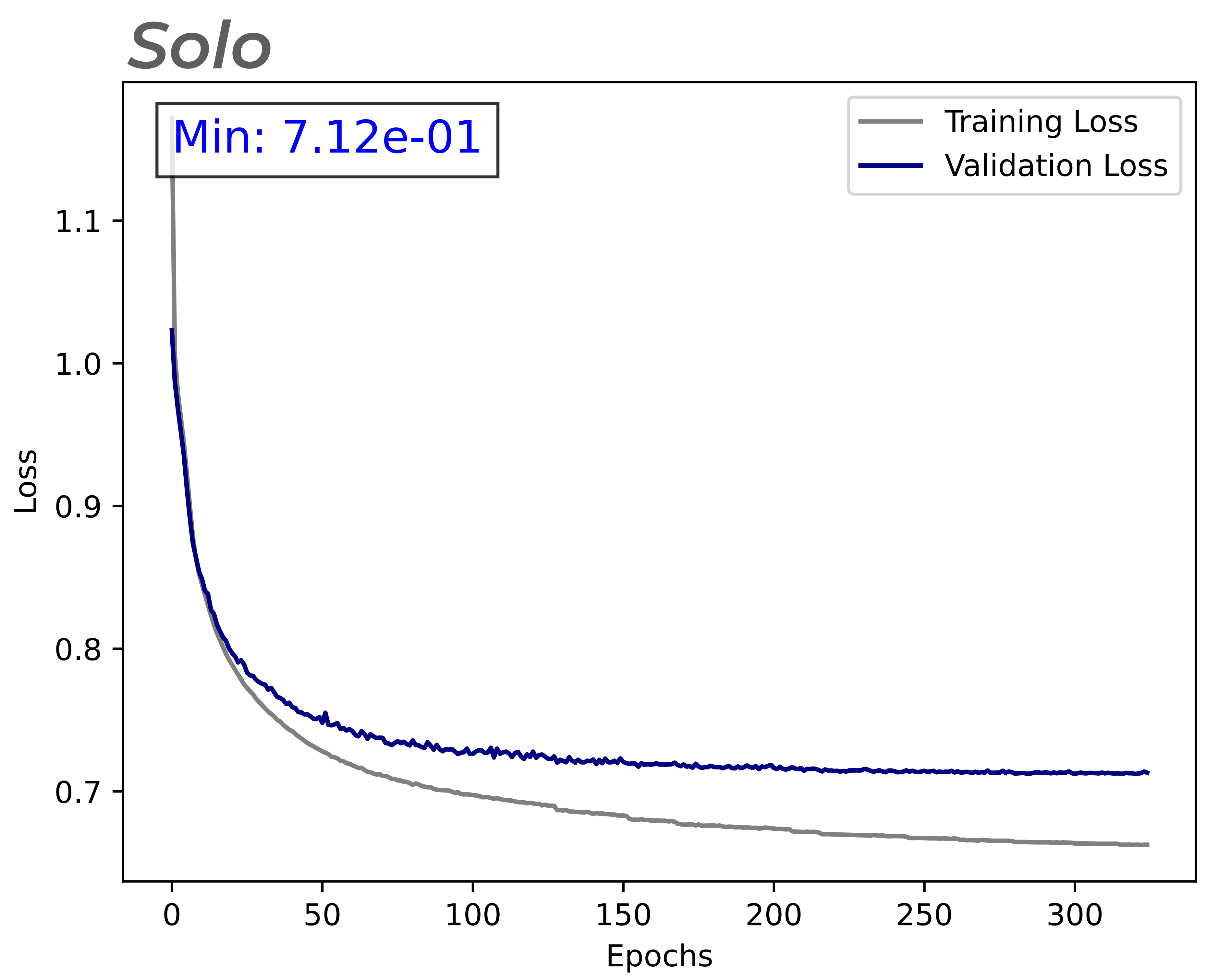
$y_T =$ 

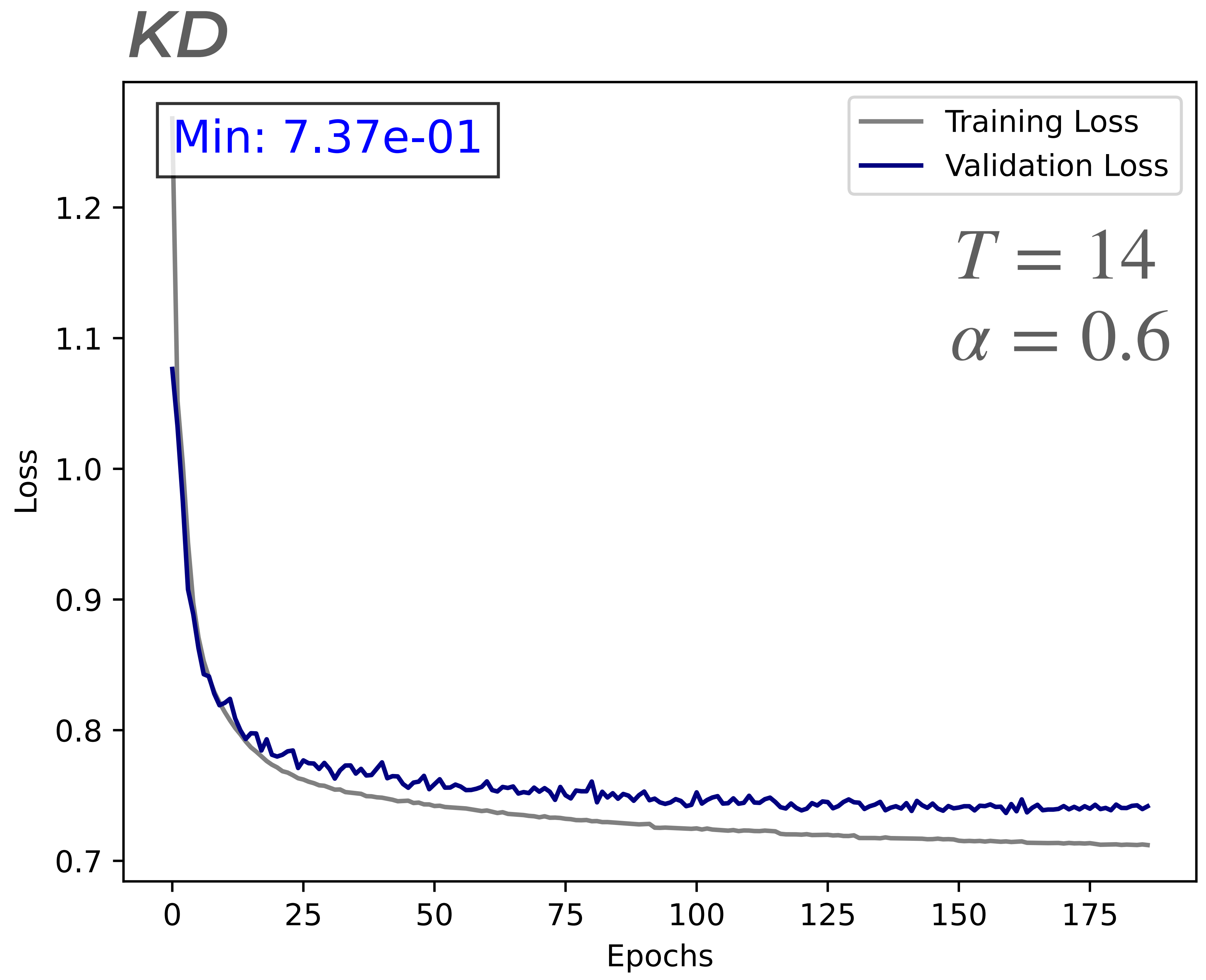
$y_S =$ 

$$\mathcal{L}(y, y_T, y_S) = \alpha \mathcal{L}_{\text{cross-entropy}}(y, y_S) + (1 - \alpha)\mathcal{L}_{\text{distillaion}}(y_T, y_S)$$

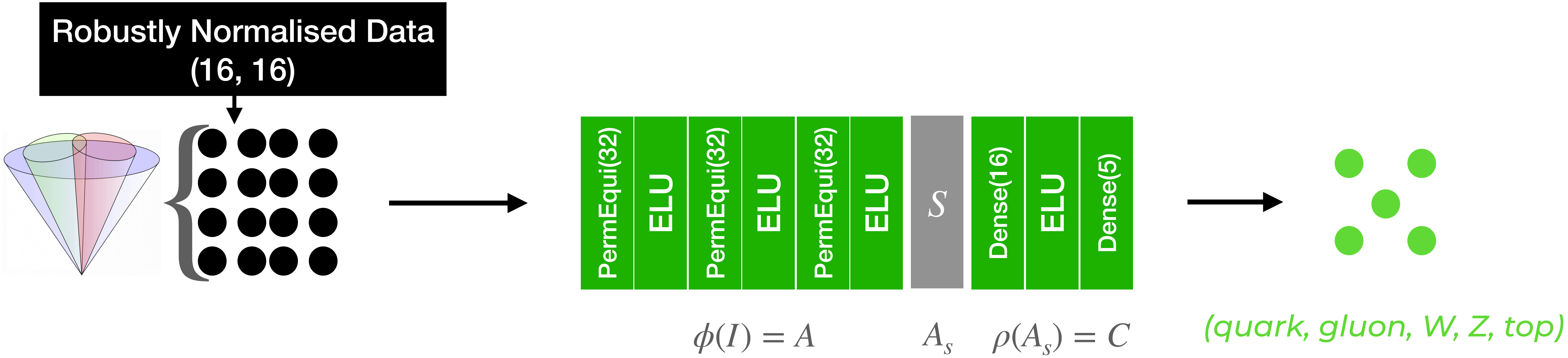

Solo
Gluon: AUC = 93.1%; FPR @ 80% TPR: 0.106
Quark: AUC = 91.0%; FPR @ 80% TPR: 0.134
W: AUC = 93.9%; FPR @ 80% TPR: 0.083
Z: AUC = 93.3%; FPR @ 80% TPR: 0.089
Top: AUC = 95.6%; FPR @ 80% TPR: 0.041
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0

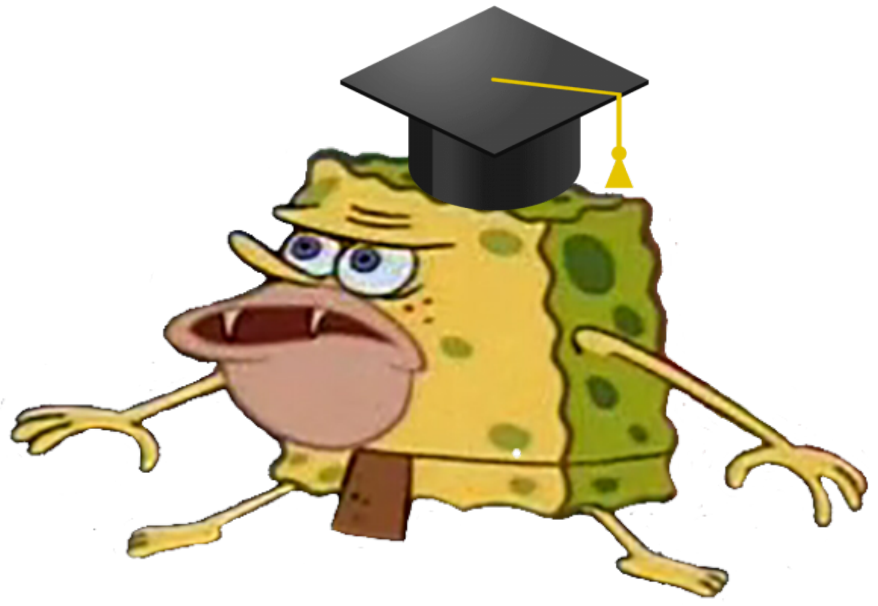

KD
Gluon: AUC = 93.1%; FPR @ 80% TPR: 0.106
Quark: AUC = 90.8%; FPR @ 80% TPR: 0.138
W: AUC = 92.8%; FPR @ 80% TPR: 0.099
Z: AUC = 92.5%; FPR @ 80% TPR: 0.109
Top: AUC = 95.5%; FPR @ 80% TPR: 0.042
$T = 14$
$\alpha = 0.6$
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0

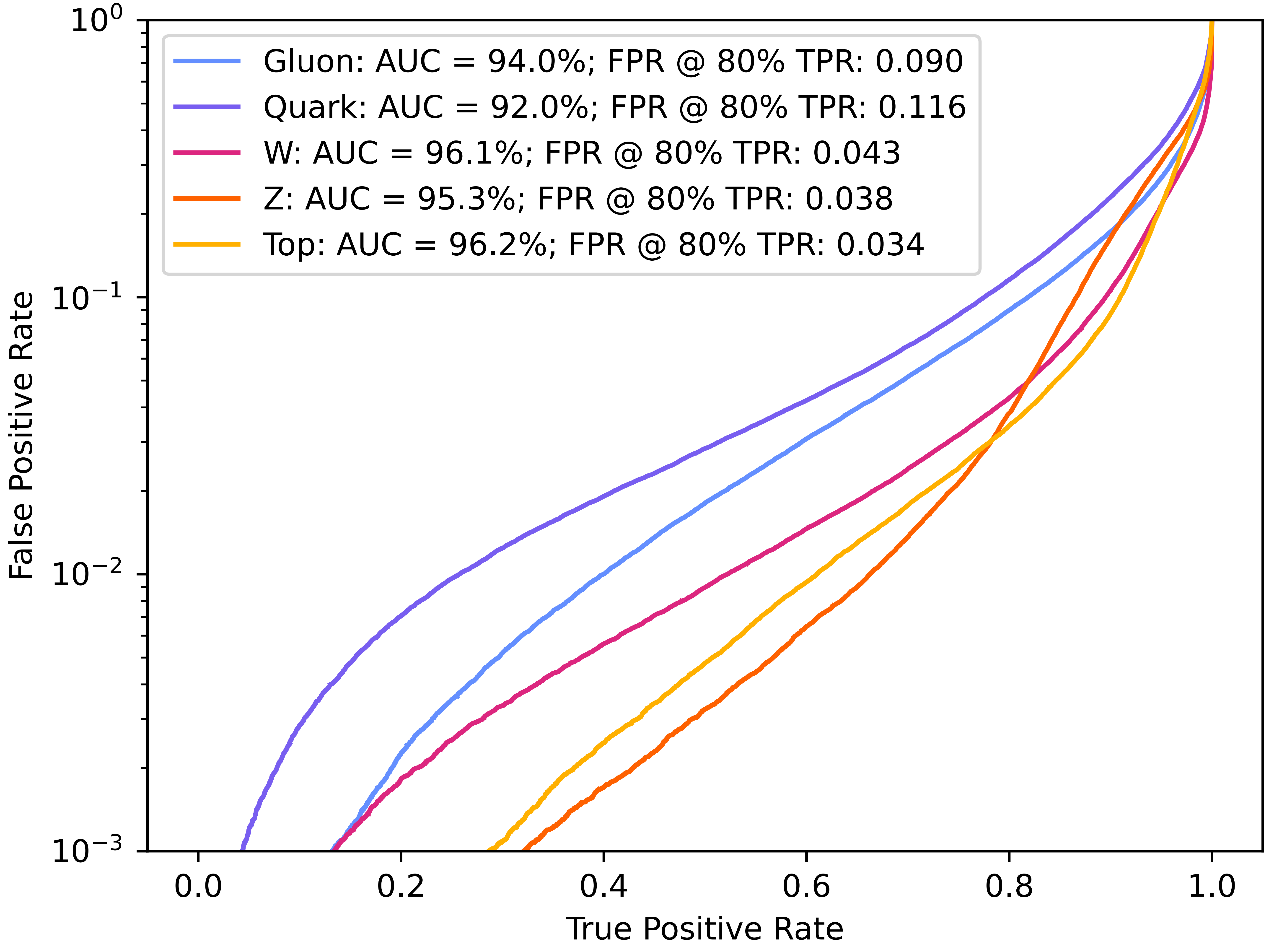

Solo
Min: 7.12e-01
Training Loss
Validation Loss
Loss
Epochs
1.1
1.0
0.9
0.8
0.7
0
50
100
150
200
250
300


KD
Min: 7.37e-01
Training Loss
Validation Loss
$T = 14$
$\alpha = 0.6$
Loss
Epochs
1.2
1.1
1.0
0.9
0.8
0.7
0
25
50
75
100
125
150
175

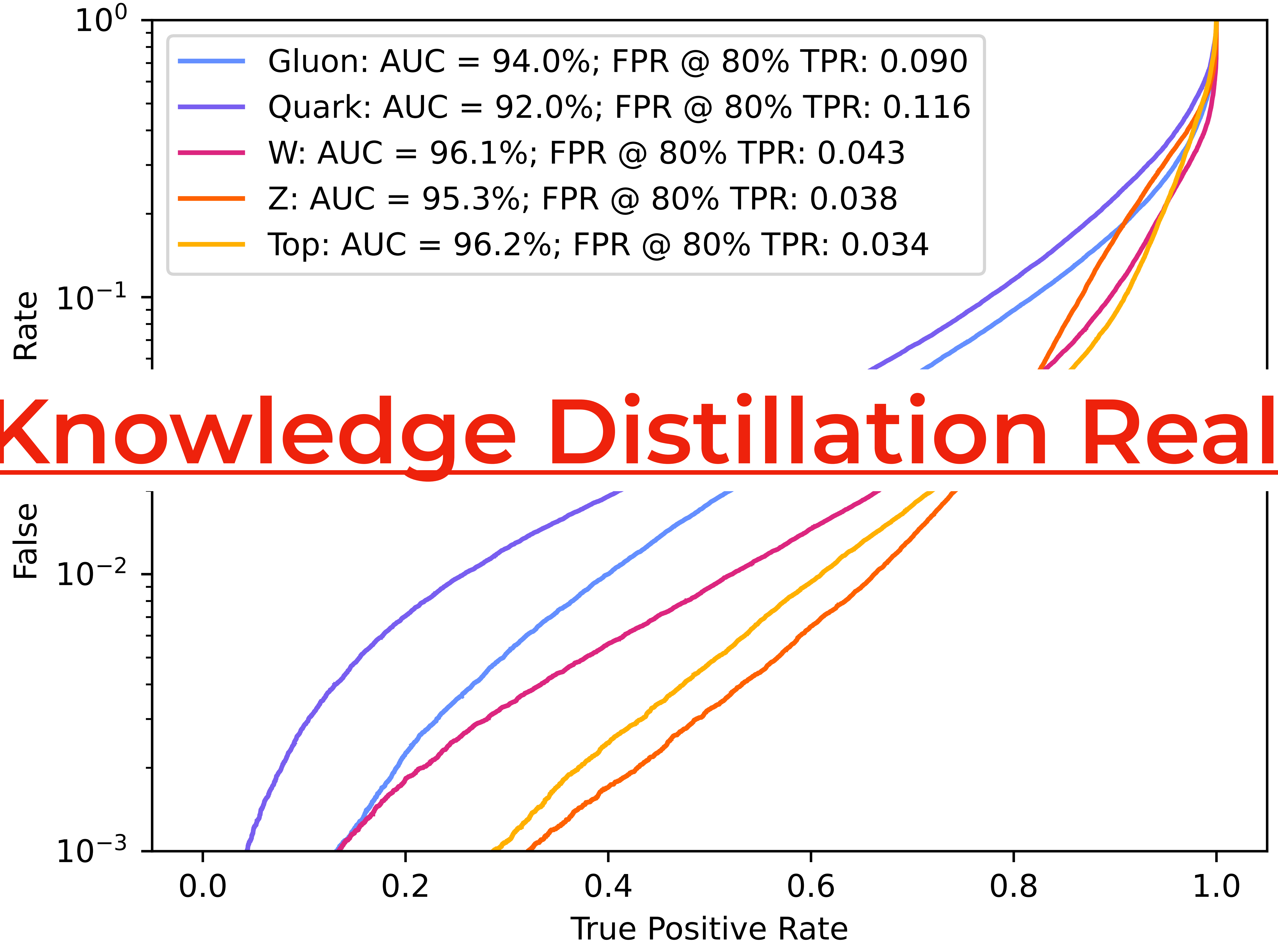
Deep Sets


Robustly Normalised Data
(16, 16)
PermEqui(32)
ELU
PermEqui(32)
ELU
PermEqui(32)
ELU
S
Dense(16)
ELU
Dense(5)
$\phi(I) = A$
A_s
$\rho(A_s) = C$
(quark, gluon, W, Z, top)




Gluon: AUC = 94.0%; FPR @ 80% TPR: 0.090
Quark: AUC = 92.0%; FPR @ 80% TPR: 0.116
W: AUC = 96.1%; FPR @ 80% TPR: 0.043
Z: AUC = 95.3%; FPR @ 80% TPR: 0.038
Top: AUC = 96.2%; FPR @ 80% TPR: 0.034
10^0
10^{-1}
10^{-2}
10^{-3}
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
True Positive Rate

Does Knowledge Distillation Really Work?



Does Knowledge Distillation Really Work?

Samuel Stanton
NYU

Pavel Izmailov
NYU

Polina Kirichenko
NYU

Alexander A. Alemi
Google Research

Andrew Gordon Wilson
NYU

https://arxiv.org/abs/2106.05945

Does Knowledge Distillation Really Work?

Samuel Stanton
NYU

Pavel Izmailov
NYU

Polina Kirichenko
NYU

Alexander A. Alemi
Google Research

Andrew Gordon Wilson
NYU

Does knowledge distillation really work? In short:

https://arxiv.org/abs/2106.05945

Does Knowledge Distillation Really Work?

Samuel Stanton
NYU

Pavel Izmailov
NYU

Polina Kirichenko
NYU

Alexander A. Alemi
Google Research

Andrew Gordon Wilson
NYU

Does knowledge distillation really work? In short: ***Yes***

https://arxiv.org/abs/2106.05945

Does Knowledge Distillation Really Work?

Samuel Stanton
NYU

Pavel Izmailov
NYU

Polina Kirichenko
NYU

Alexander A. Alemi
Google Research

Andrew Gordon Wilson
NYU

Does knowledge distillation really work? In short: *Yes*, in the sense that it often improves student generalization. *No*, in that knowledge distillation often fails to live up to its name, transferring very limited knowledge from teacher to student.

https://arxiv.org/abs/2106.05945

Moral of the Story

- KD is a good paradigm for model compression
- Ensemble knowledge distillation: safe
- Distilling across architectures: caution
- Distilling a big model to a smaller version of itself: depends on dataset
- You can use KD for regularisation.

Model Compression - Bucila et. al.

Distilling Knowledge in a Neural Network - Hinton et. al.

Knowledge Distillation: A Survey - Gou et. al.

Does Knowledge Distillation Really Work? - Stanton et. al.

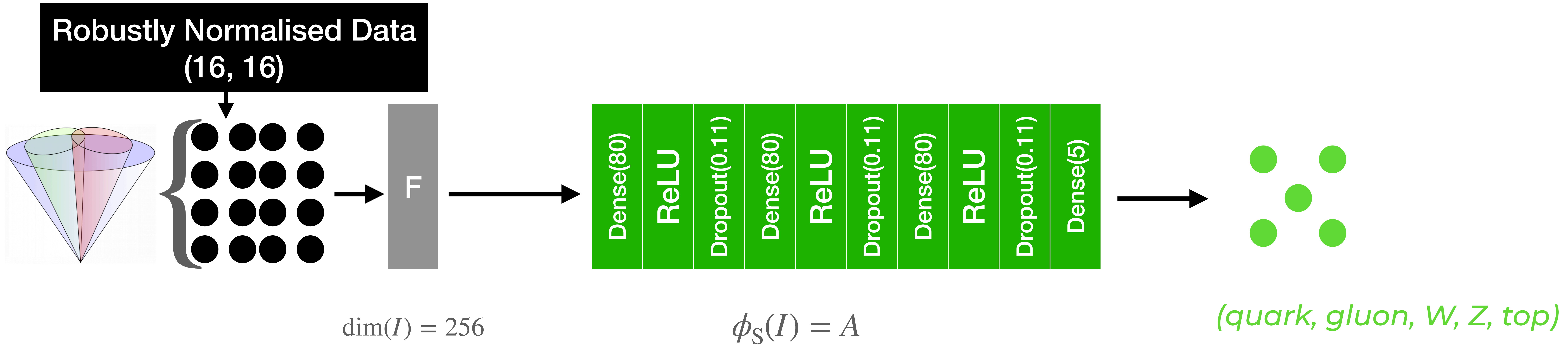
FitNets: Hints for Thin Deep Jets - Stanton et. al.

https://keras.io/examples/vision/knowledge_distillation/

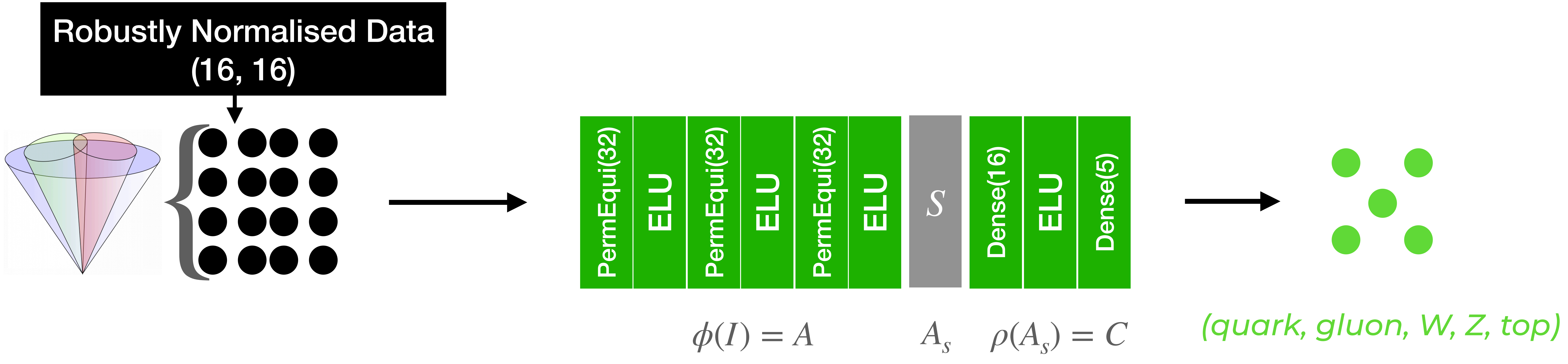
backup

JEDInet DNN

arxiv:1908.05318v3



Deep Sets



Permutation Equivariant Layer + Activation

$$f(\mathbf{x}) = \sigma(\mathbf{x}\Lambda - \mathbf{1}\mathbf{1}^\top \mathbf{x}\Gamma)$$

$$f(\mathbf{x}) = \sigma(\mathbf{x}\Lambda - \mathbf{1}\,\mathrm{maxpool}(\mathbf{x})\Gamma)$$

Non-linearity applied to weighted combination of

➡ Input.

➡ Sum of output values.

$$\mathrm{H}_q = -\,\mathbb{E}_{x \sim p(\cdot)}[\log q(x)]$$

$$\frac{\partial H}{\partial g(z)_i} \approx \frac{1}{T}\left(\frac{1 + g(z)_i/T}{N + \sum_j g(z)_j/T} - \frac{1 + f(z)_i/T}{N + \sum_j f(z)_j/T}\right) \quad T \gg 1$$

$$\frac{\partial H}{\partial g(z)} \approx \frac{1}{NT^2}(g(z) - f(z))$$

model compression!

http://arxiv.org/abs/1503.02531

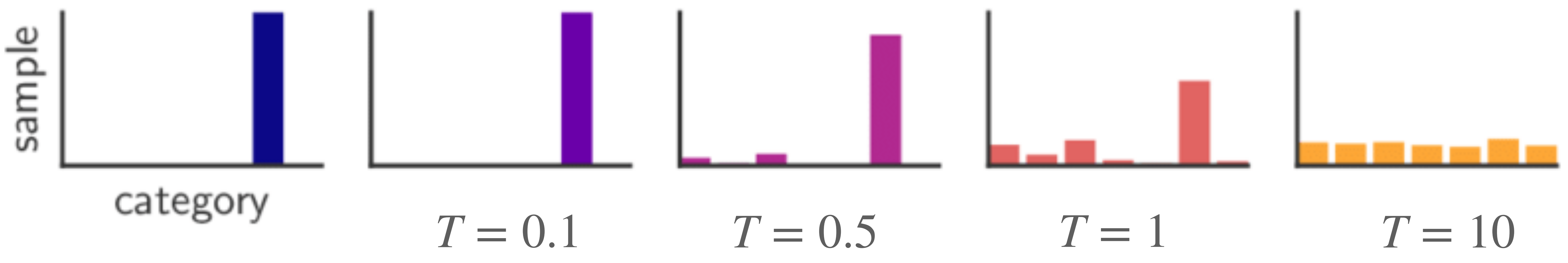
$$\mathrm{H}_q = -\,\mathbb{E}_{z\sim p(\cdot)}[\log q(z)]$$

$$\begin{aligned}
\frac{\partial C}{\partial z_i} &= -\sum_j p_j \frac{\partial \log(q_j)}{\partial z_i} = -\sum_j \frac{p_j}{q_j}\frac{\partial q_j}{\partial z_i} = -\frac{p_i}{q_i}\frac{\partial q_i}{\partial z_i} - \sum_{j\neq i}\frac{p_j}{q_j}\frac{\partial q_j}{\partial z_i} \\
&= -\frac{1}{T}\frac{p_i}{q_i}q_i(1-q_i) - \sum_{j\neq i}\frac{1}{T}\frac{p_j}{q_j}(-q_j q_i) \\
&= -\frac{p_i}{T} + \frac{1}{T}p_i q_i + \sum_{j\neq i} p_j q_i \\
&= -\frac{p_i}{T} + \frac{q_i}{T}\sum_j p_j = \frac{q_i - p_i}{T} \\
&= \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right)
\end{aligned}$$

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1+z_i/T}{N+\sum_j z_j/T} - \frac{1+v_i/T}{N+\sum_j v_j/T}\right)$$

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i)$$

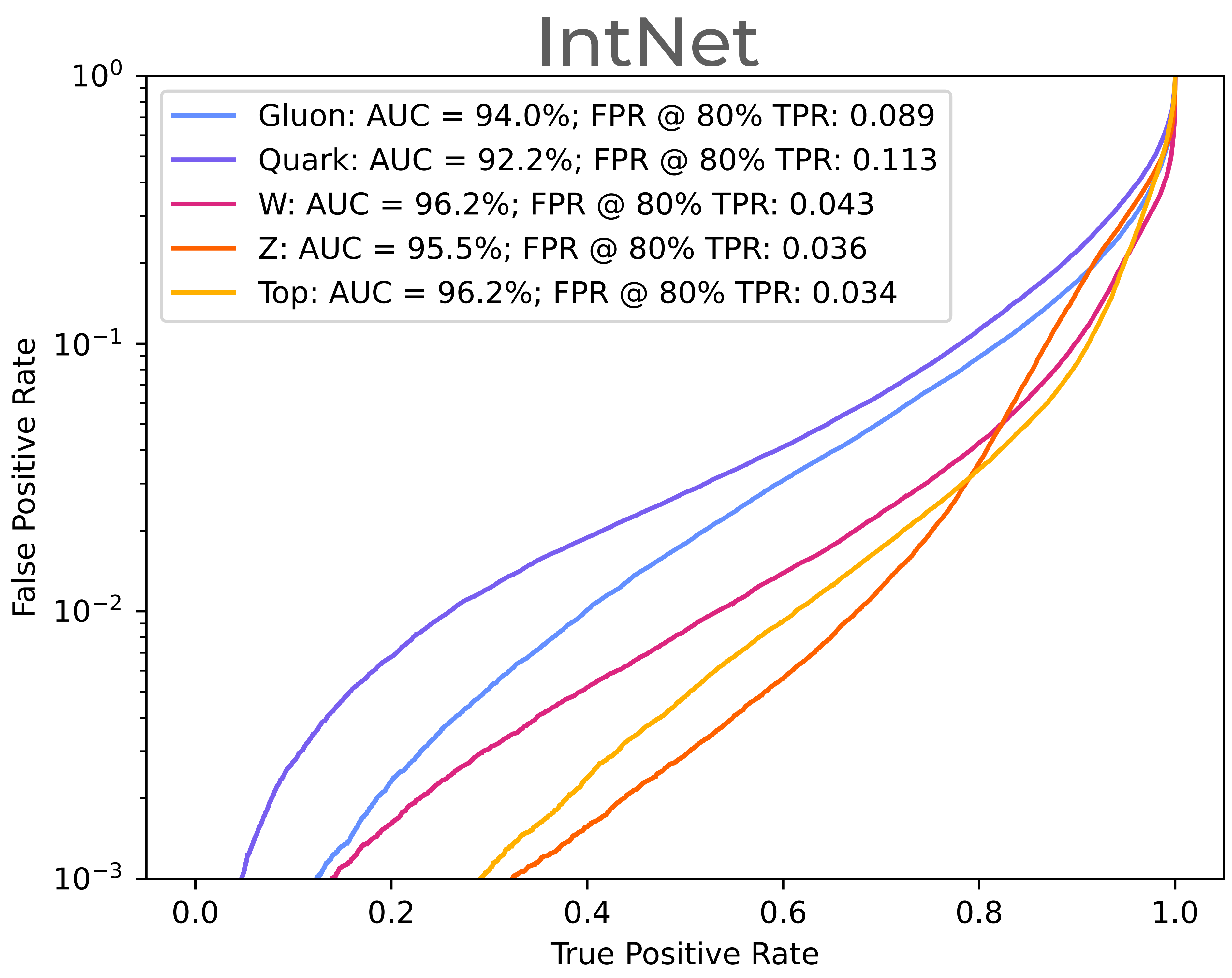
http://arxiv.org/abs/1503.02531

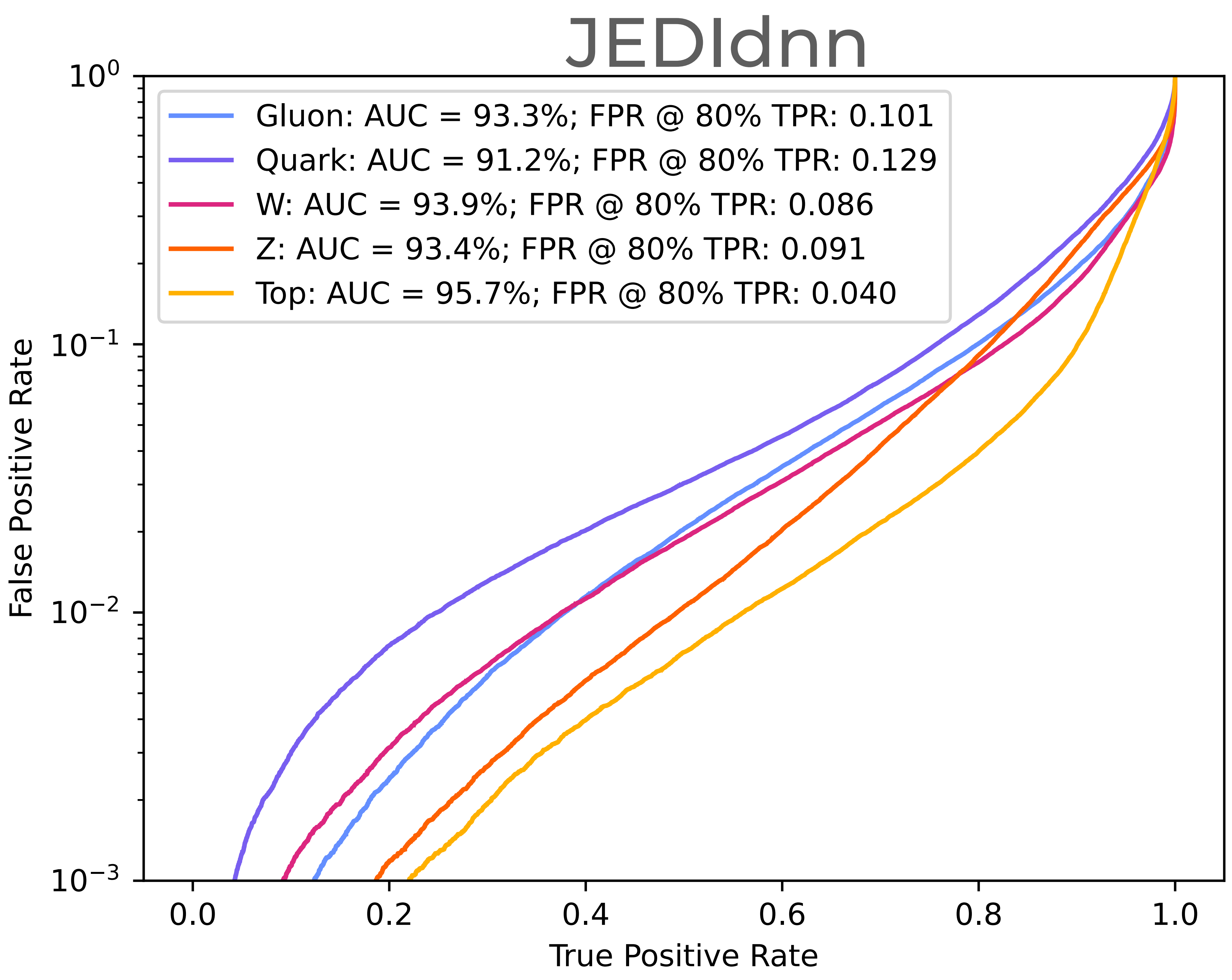


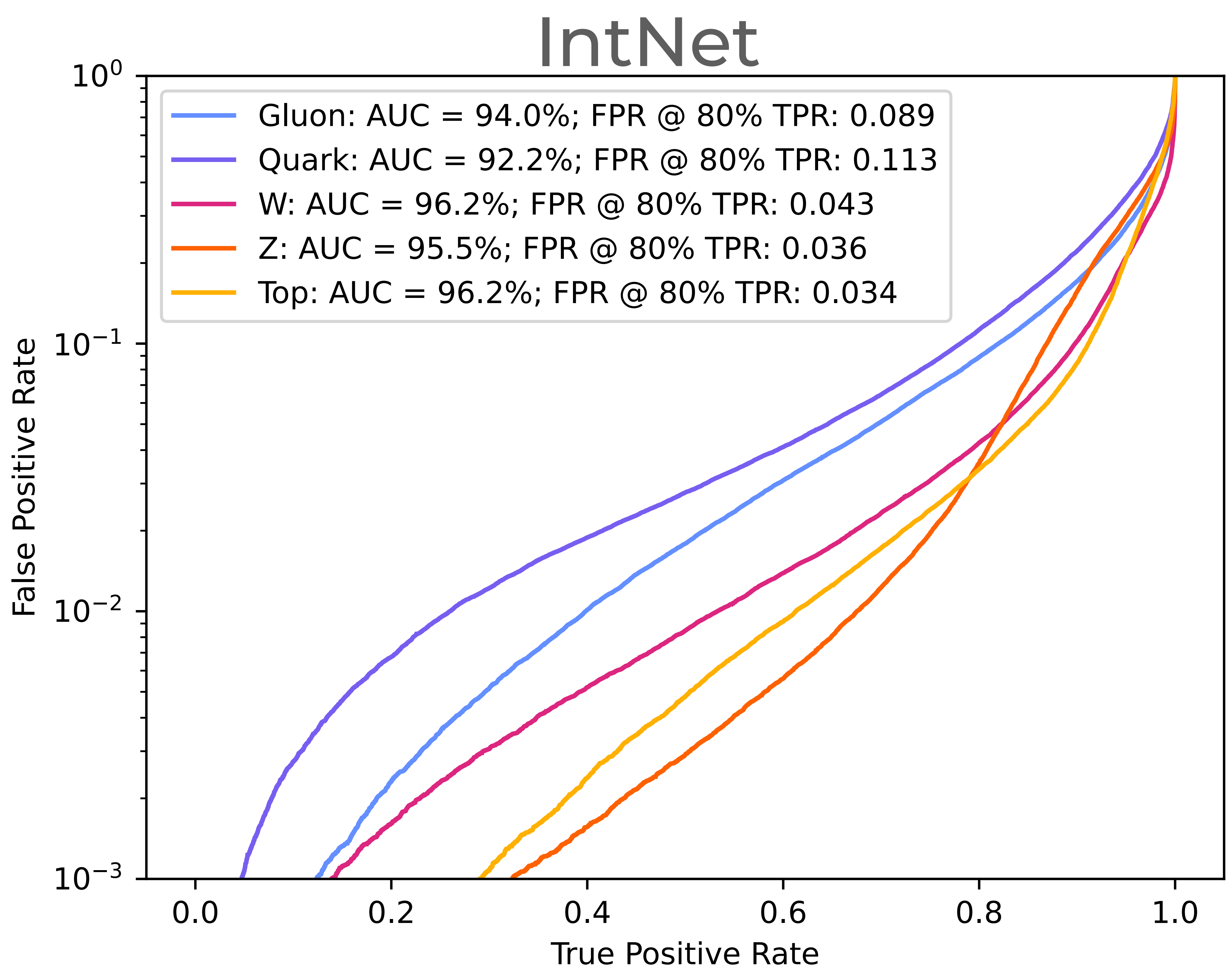
➡ Low temperature leads to harder labels.

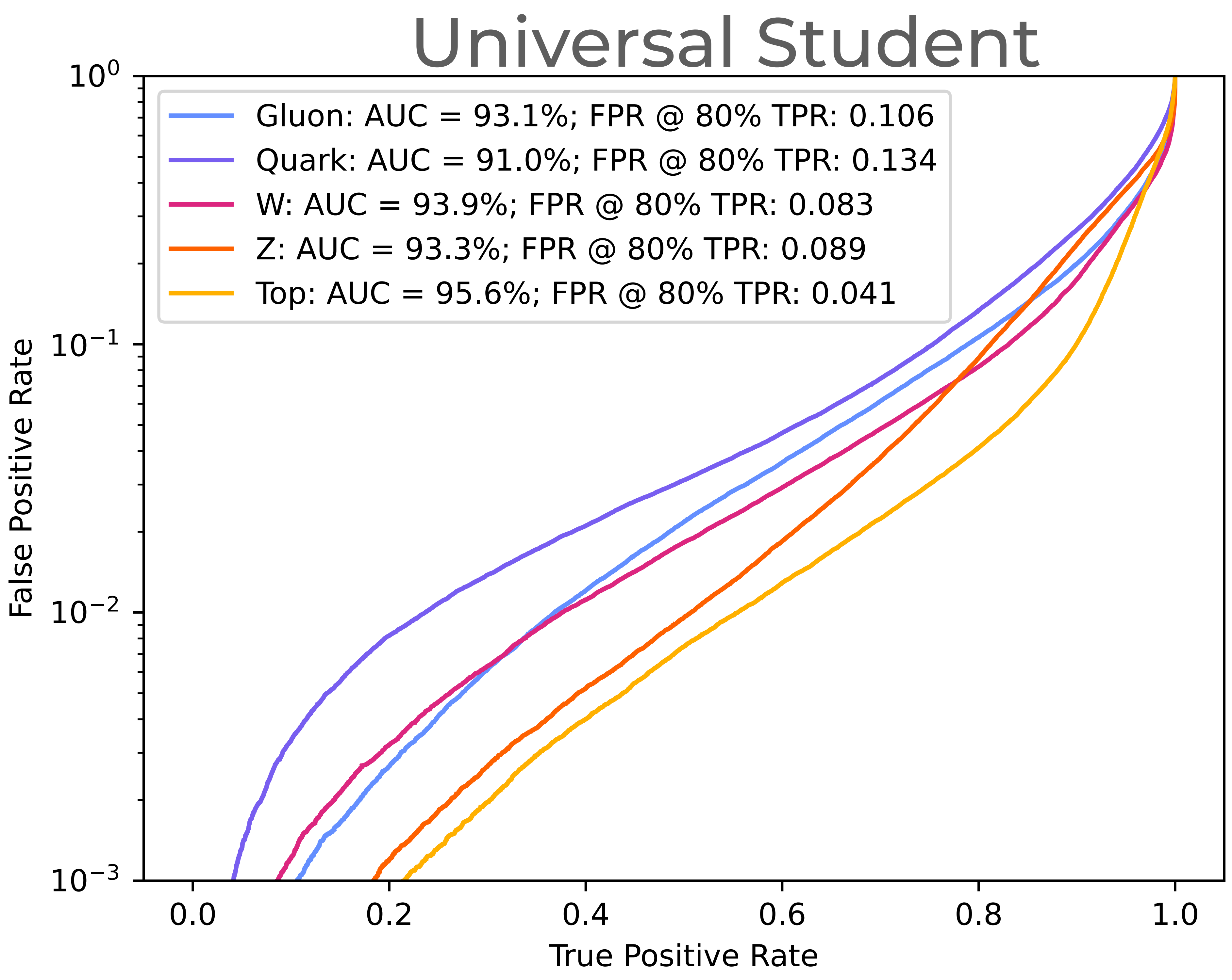
➡ High temperature leads to softer labels.

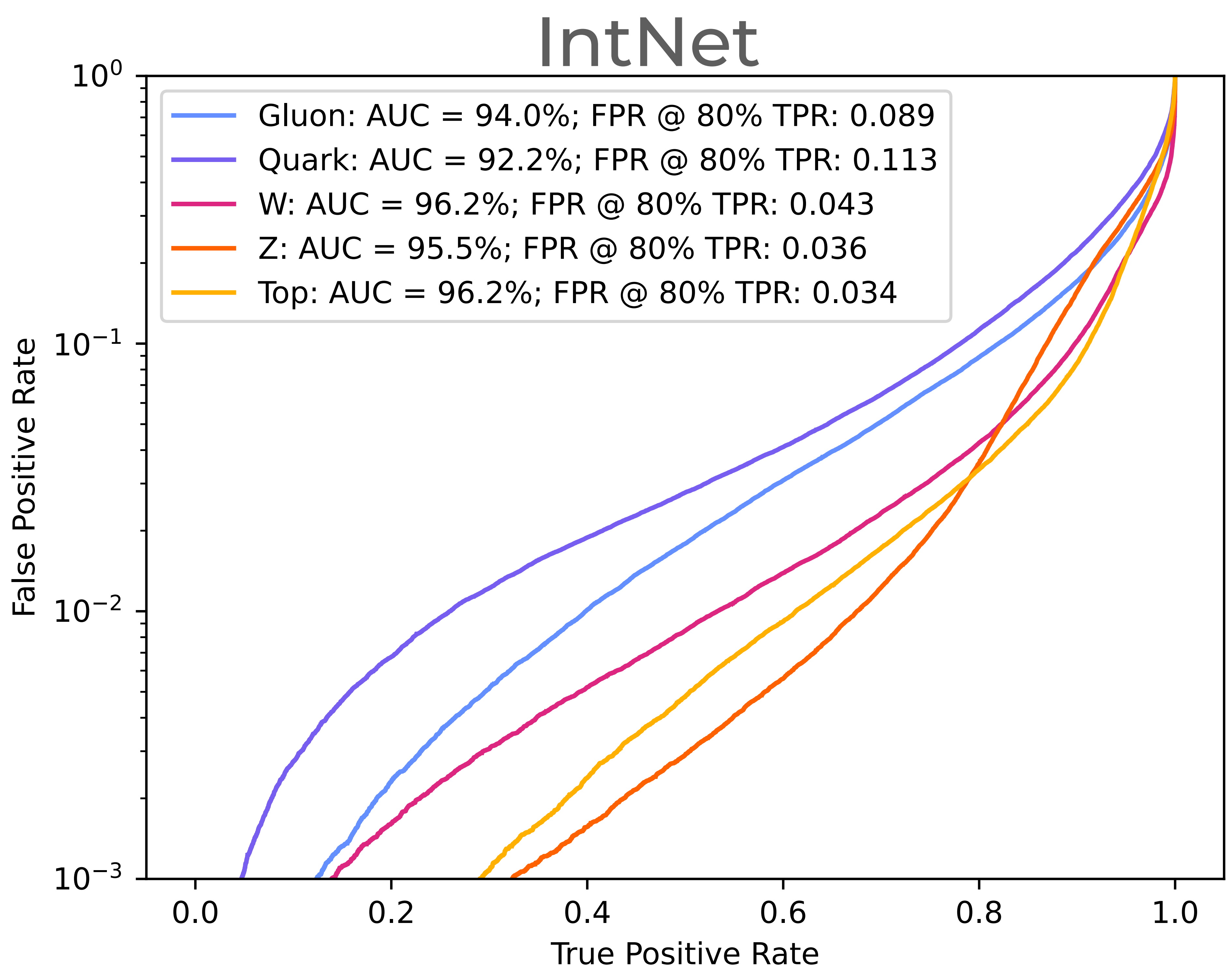
➡ For some (maybe all? - proven only for categorical cross-entropy) loss functions, very high temperatures lead to equivalency between knowledge distillation and logit matching (MSE between teacher and student logits).

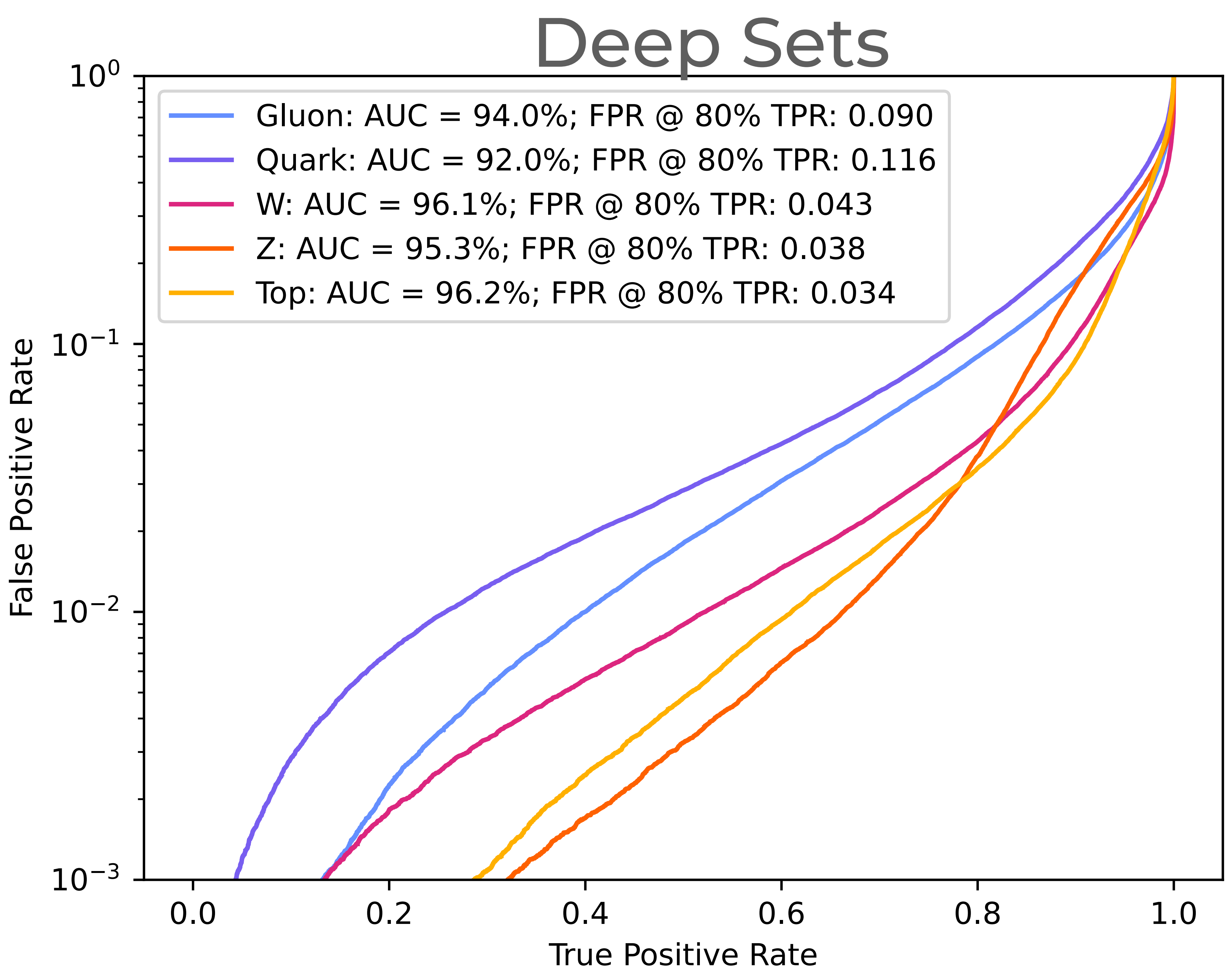

IntNet
Gluon: AUC = 94.0%; FPR @ 80% TPR: 0.089
Quark: AUC = 92.2%; FPR @ 80% TPR: 0.113
W: AUC = 96.2%; FPR @ 80% TPR: 0.043
Z: AUC = 95.5%; FPR @ 80% TPR: 0.036
Top: AUC = 96.2%; FPR @ 80% TPR: 0.034
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0


JEDIdnn
Gluon: AUC = 93.3%; FPR @ 80% TPR: 0.101
Quark: AUC = 91.2%; FPR @ 80% TPR: 0.129
W: AUC = 93.9%; FPR @ 80% TPR: 0.086
Z: AUC = 93.4%; FPR @ 80% TPR: 0.091
Top: AUC = 95.7%; FPR @ 80% TPR: 0.040
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0


IntNet
Gluon: AUC = 94.0%; FPR @ 80% TPR: 0.089
Quark: AUC = 92.2%; FPR @ 80% TPR: 0.113
W: AUC = 96.2%; FPR @ 80% TPR: 0.043
Z: AUC = 95.5%; FPR @ 80% TPR: 0.036
Top: AUC = 96.2%; FPR @ 80% TPR: 0.034
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0


Universal Student
Gluon: AUC = 93.1%; FPR @ 80% TPR: 0.106
Quark: AUC = 91.0%; FPR @ 80% TPR: 0.134
W: AUC = 93.9%; FPR @ 80% TPR: 0.083
Z: AUC = 93.3%; FPR @ 80% TPR: 0.089
Top: AUC = 95.6%; FPR @ 80% TPR: 0.041
False Positive Rate
True Positive Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.2
0.4
0.6
0.8
1.0

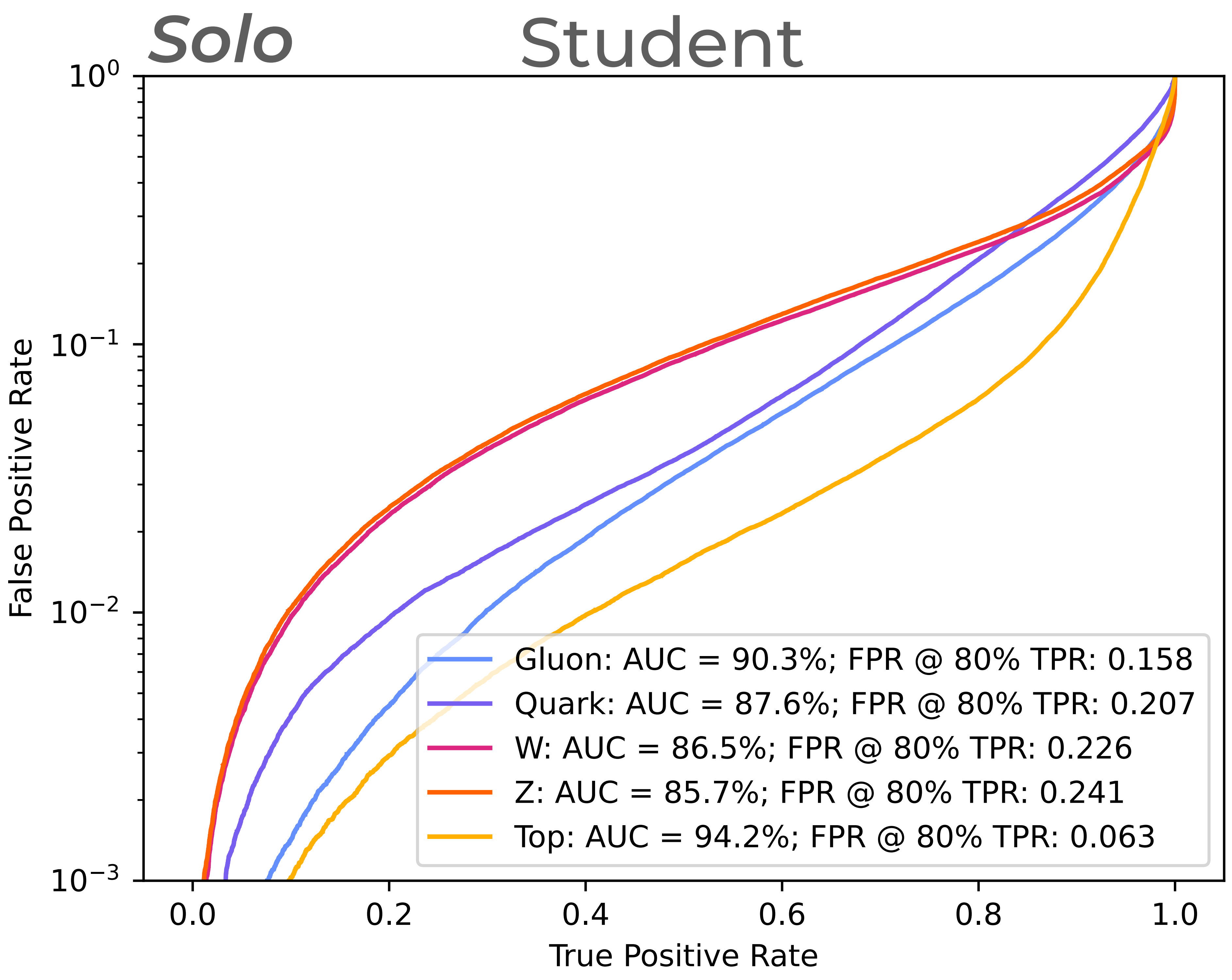

IntNet
Gluon: AUC = 94.0%; FPR @ 80% TPR: 0.089
Quark: AUC = 92.2%; FPR @ 80% TPR: 0.113
W: AUC = 96.2%; FPR @ 80% TPR: 0.043
Z: AUC = 95.5%; FPR @ 80% TPR: 0.036
Top: AUC = 96.2%; FPR @ 80% TPR: 0.034
False Positive Rate
True Positive Rate

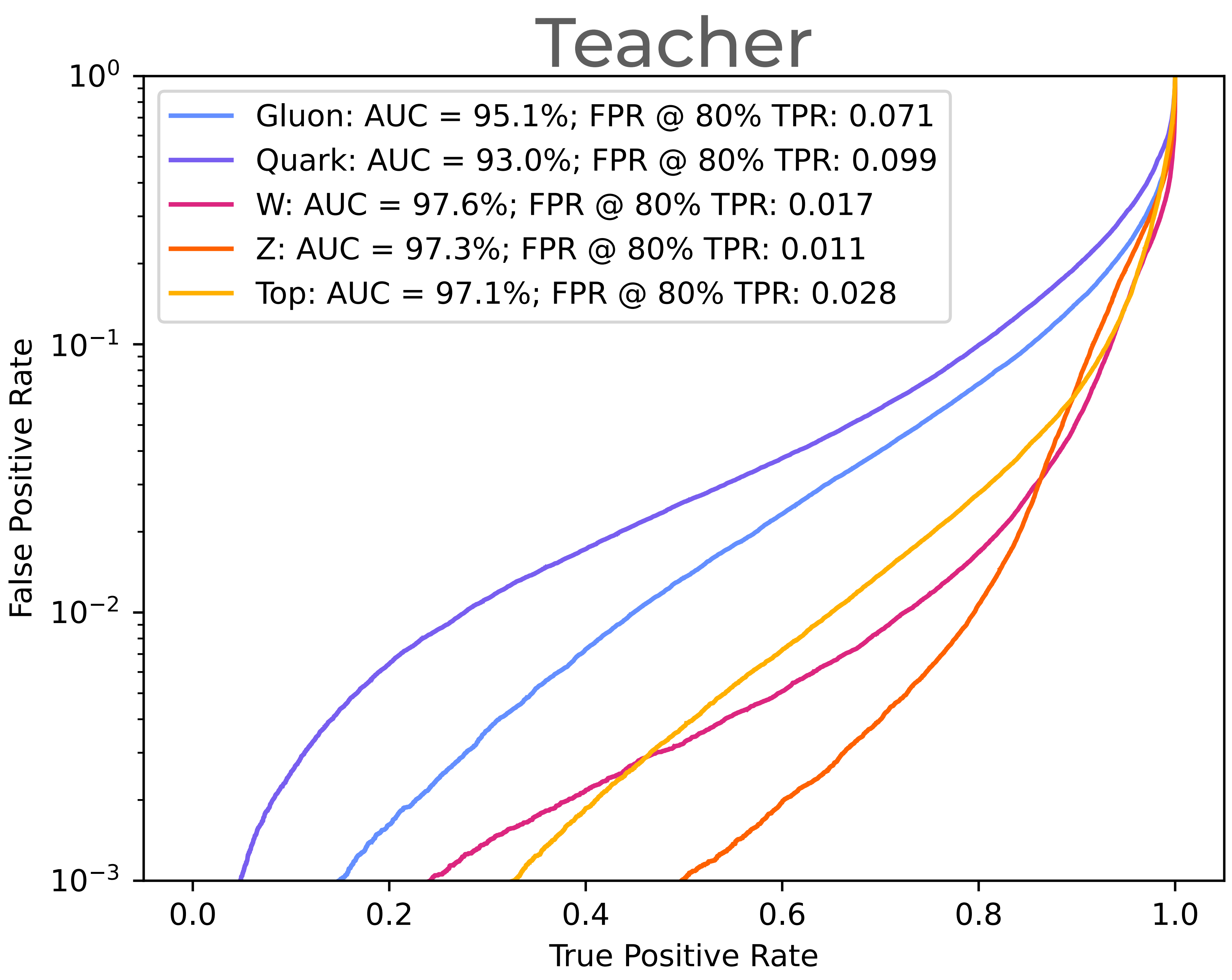

Deep Sets
Gluon: AUC = 94.0%; FPR @ 80% TPR: 0.090
Quark: AUC = 92.0%; FPR @ 80% TPR: 0.116
W: AUC = 96.1%; FPR @ 80% TPR: 0.043
Z: AUC = 95.3%; FPR @ 80% TPR: 0.038
Top: AUC = 96.2%; FPR @ 80% TPR: 0.034
False Positive Rate
True Positive Rate

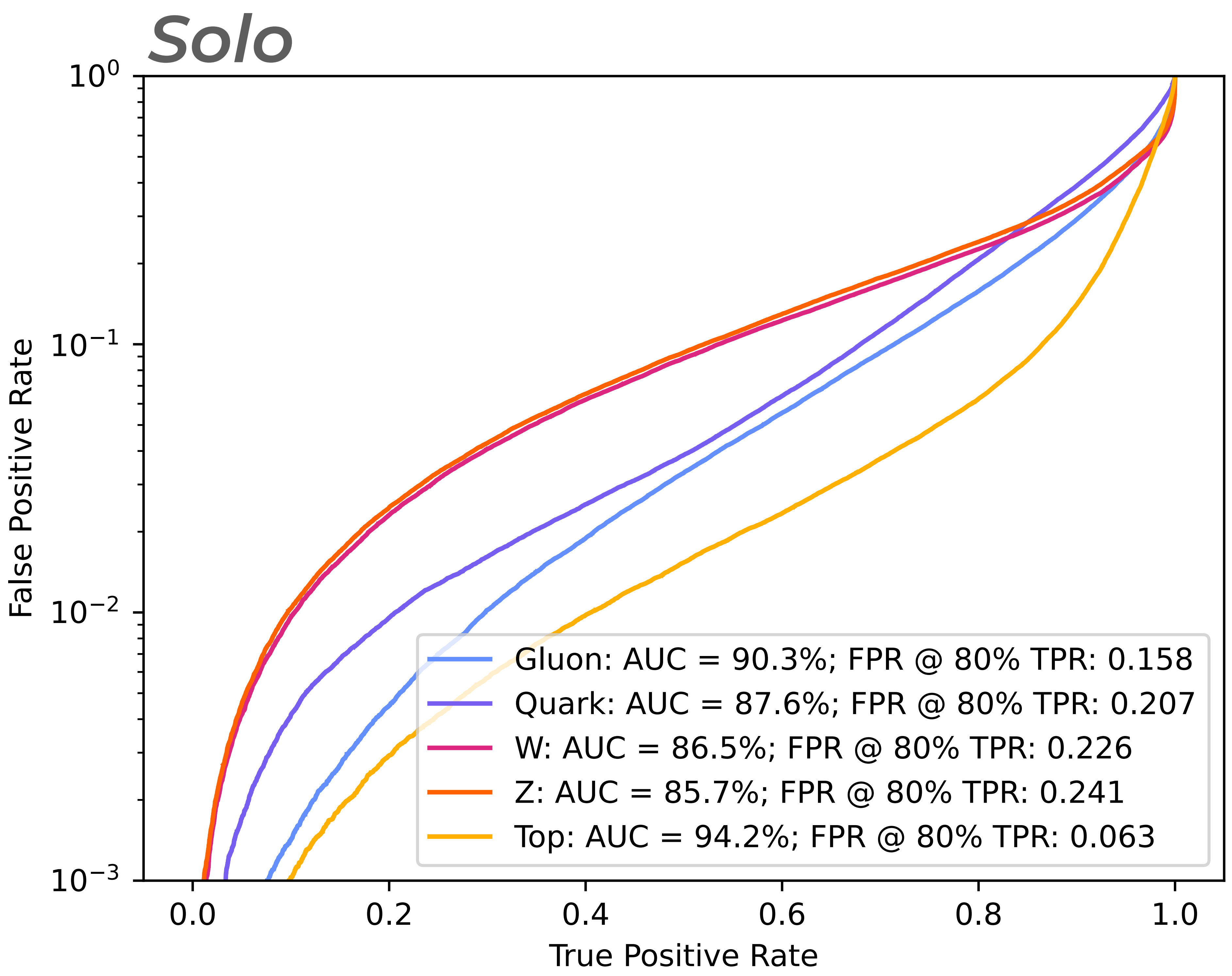
JEDInet DNN

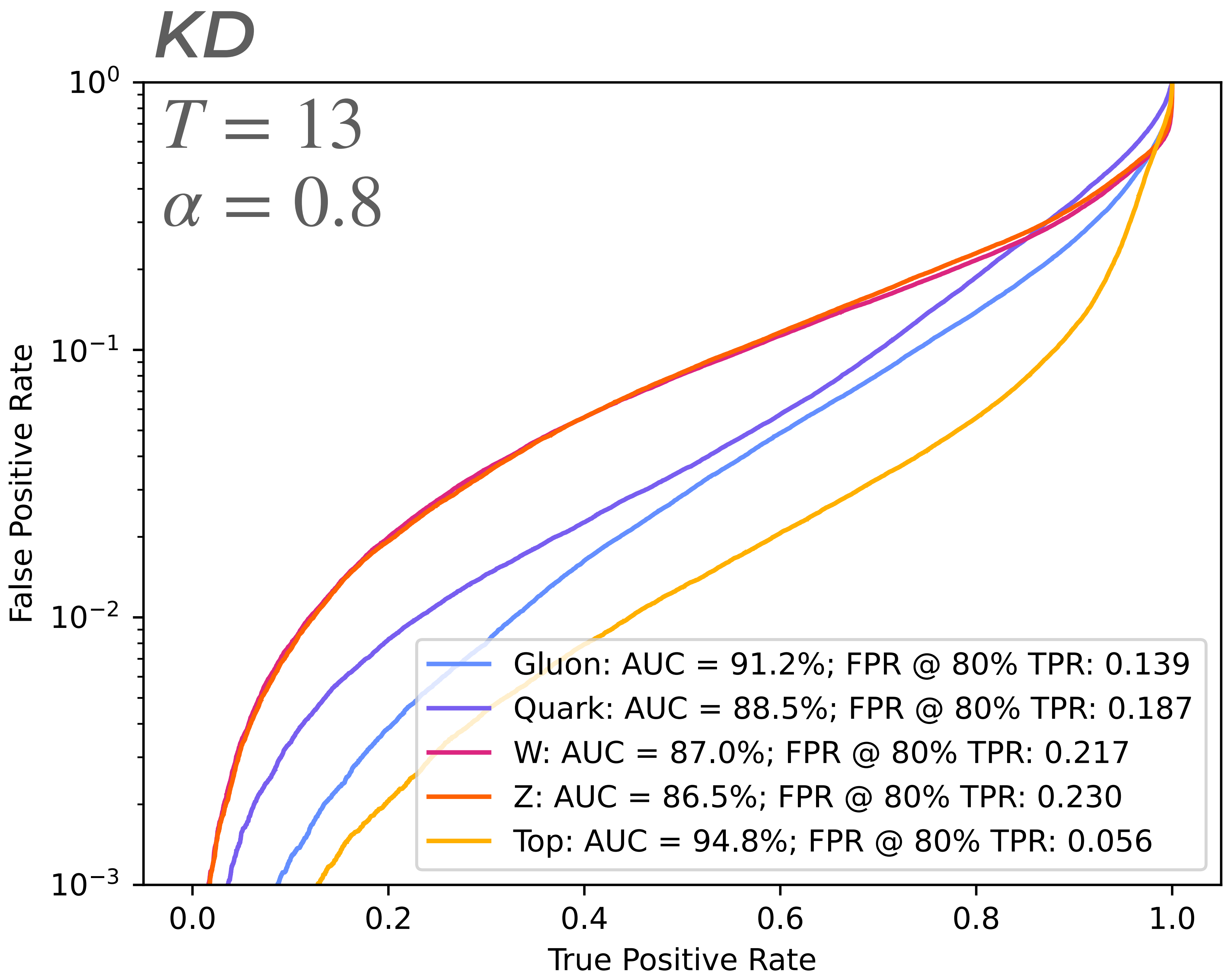
arxiv:1908.05318v3

150 constituents

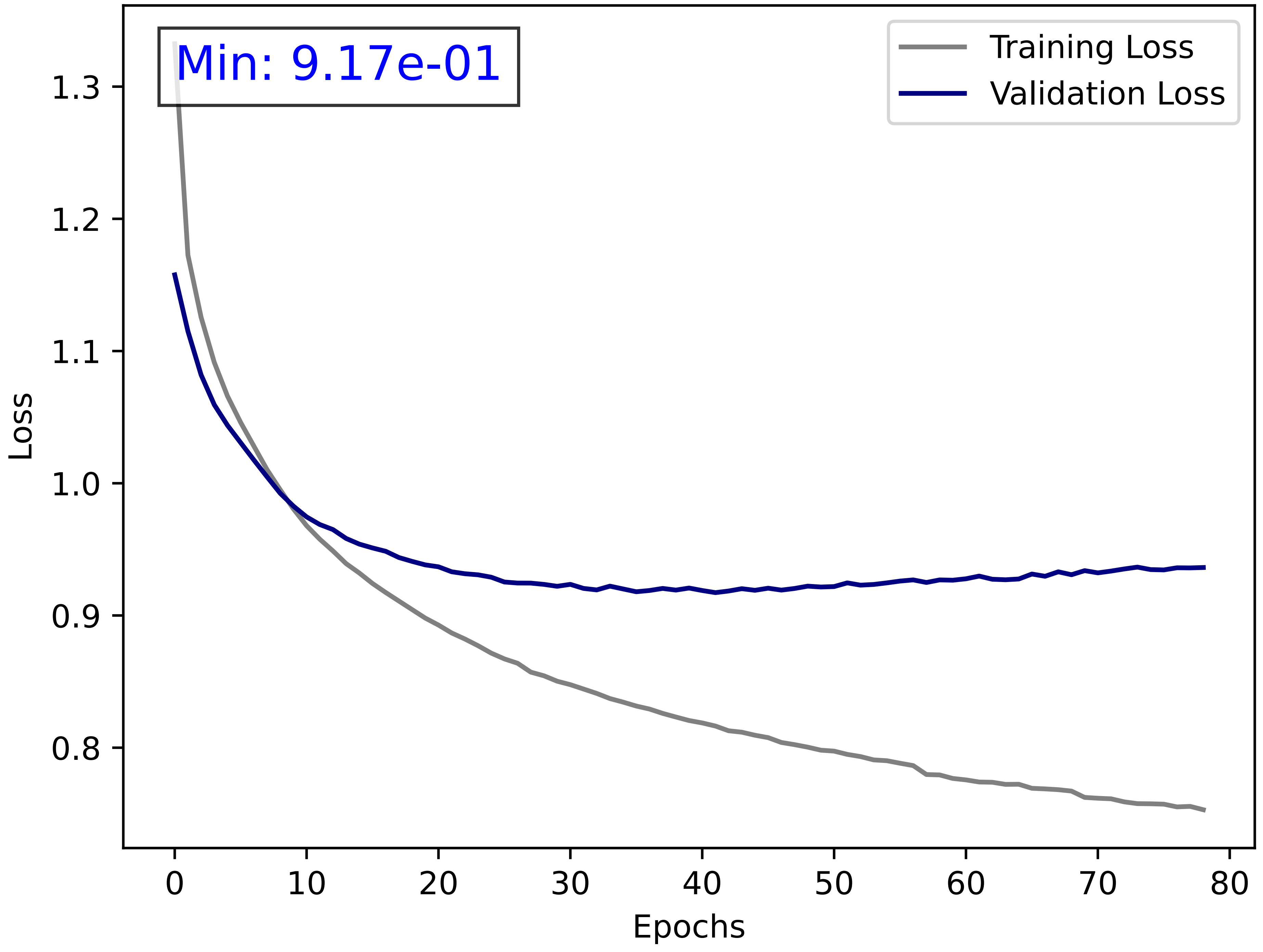





Solo
False Positive Rate
True Positive Rate
Gluon: AUC = 90.3%; FPR @ 80% TPR: 0.158
Quark: AUC = 87.6%; FPR @ 80% TPR: 0.207
W: AUC = 86.5%; FPR @ 80% TPR: 0.226
Z: AUC = 85.7%; FPR @ 80% TPR: 0.241
Top: AUC = 94.2%; FPR @ 80% TPR: 0.063

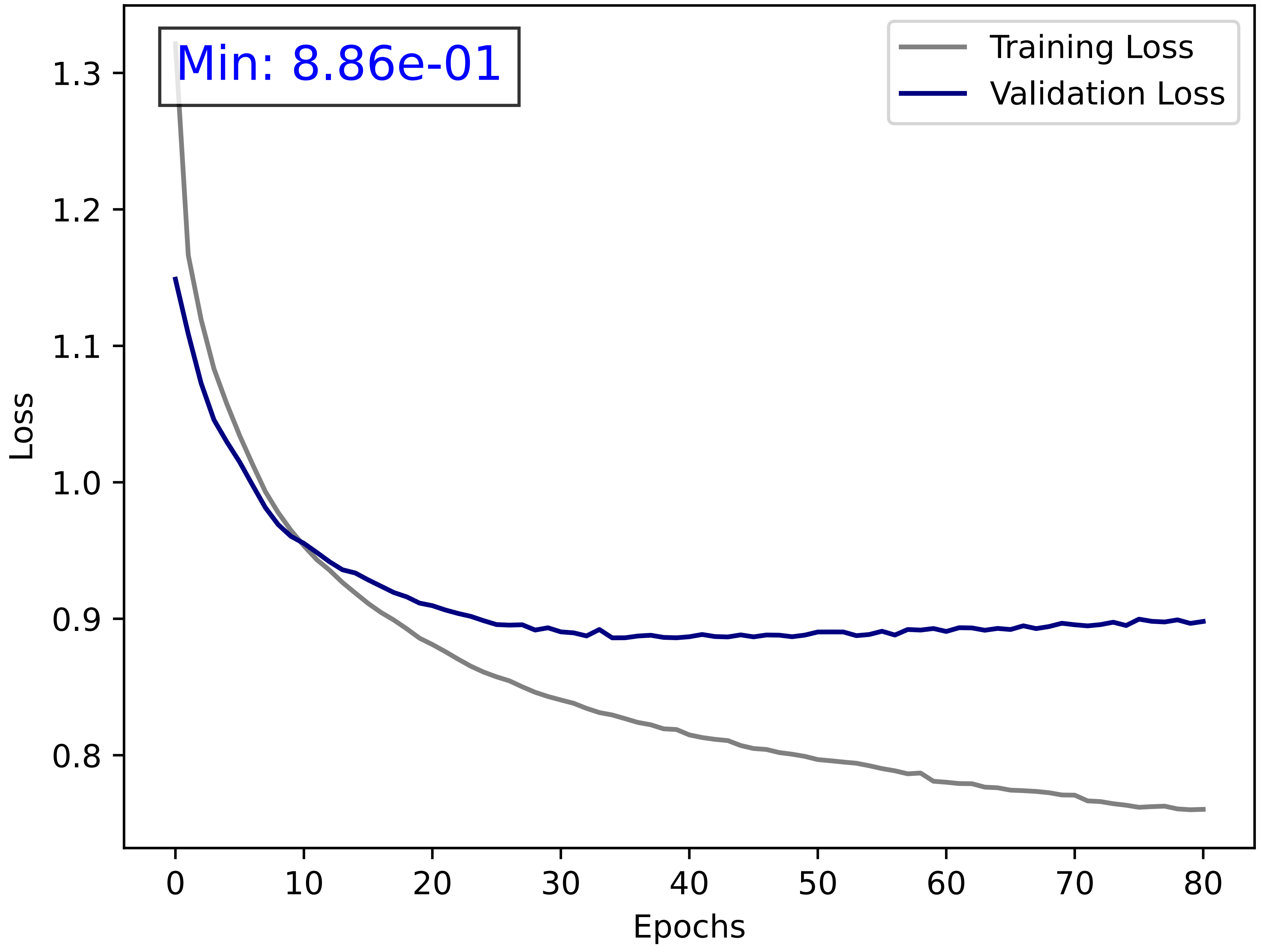

KD
$T = 13$
$\alpha = 0.8$
False Positive Rate
True Positive Rate
Gluon: AUC = 91.2%; FPR @ 80% TPR: 0.139
Quark: AUC = 88.5%; FPR @ 80% TPR: 0.187
W: AUC = 87.0%; FPR @ 80% TPR: 0.217
Z: AUC = 86.5%; FPR @ 80% TPR: 0.230
Top: AUC = 94.8%; FPR @ 80% TPR: 0.056

Solo



Min: 9.17e-01
Training Loss
Validation Loss
Loss
Epochs

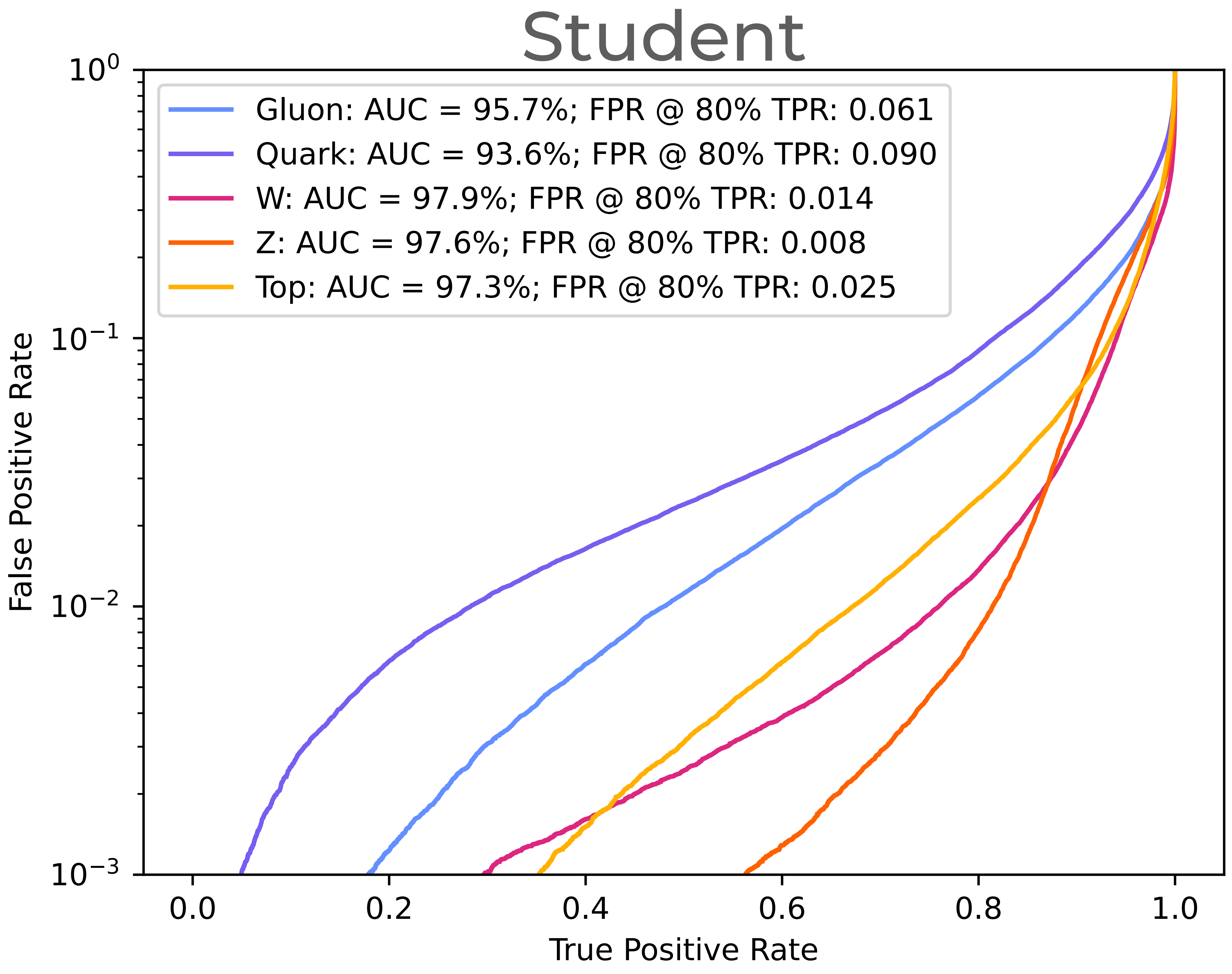
KD

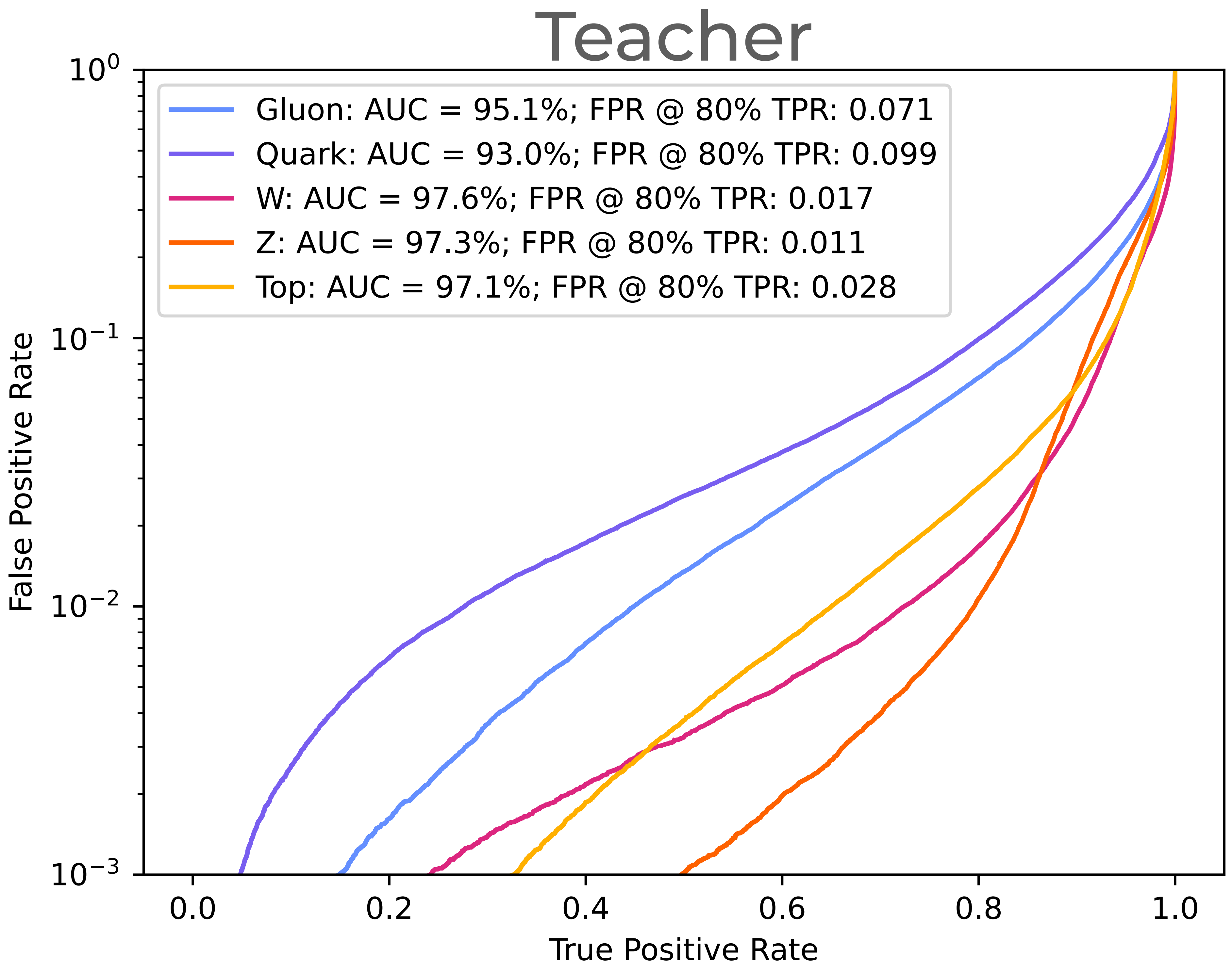


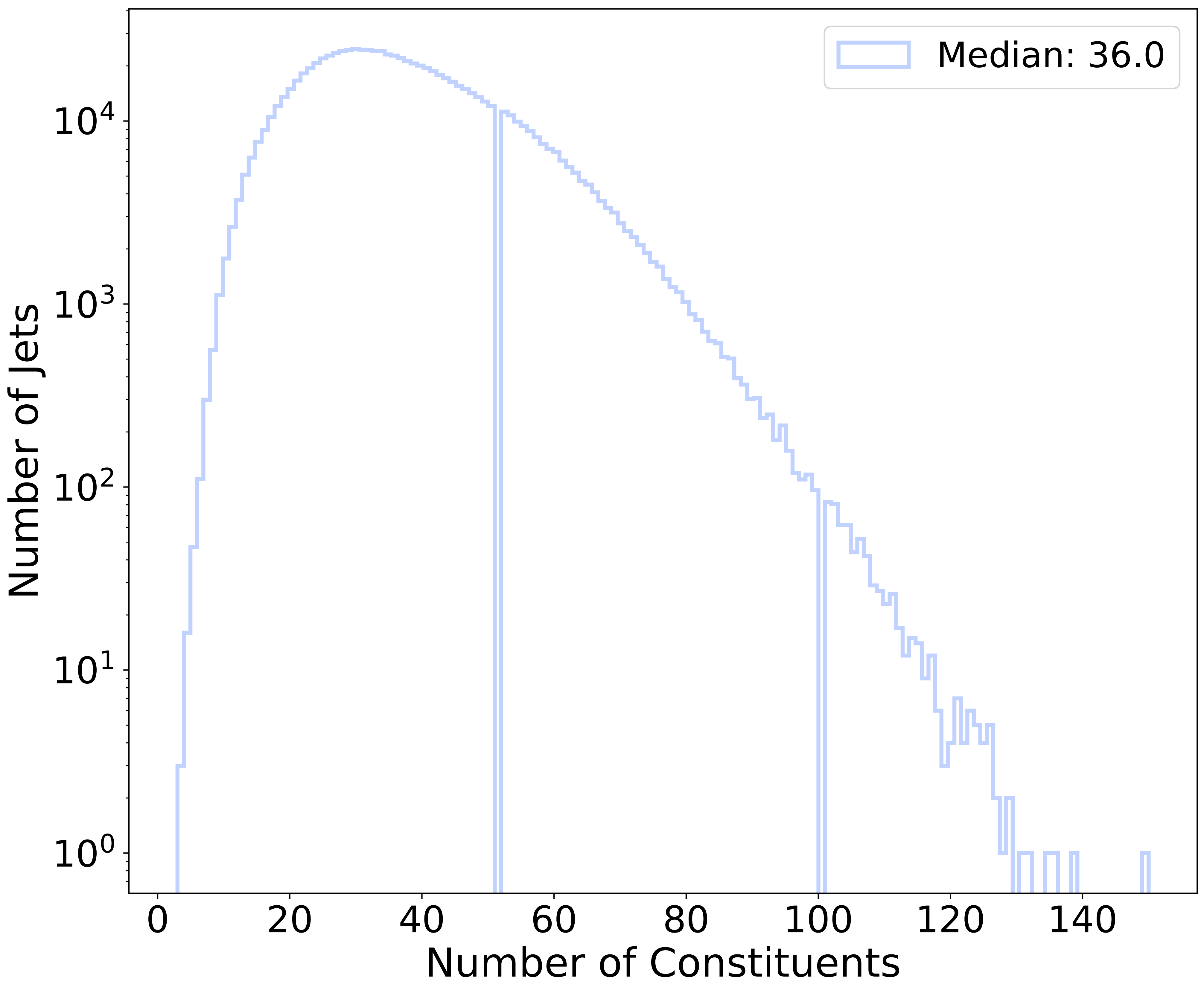
Min: 8.86e-01
Training Loss
Validation Loss
Loss
Epochs

Deep Sets

150 constituents 16 features



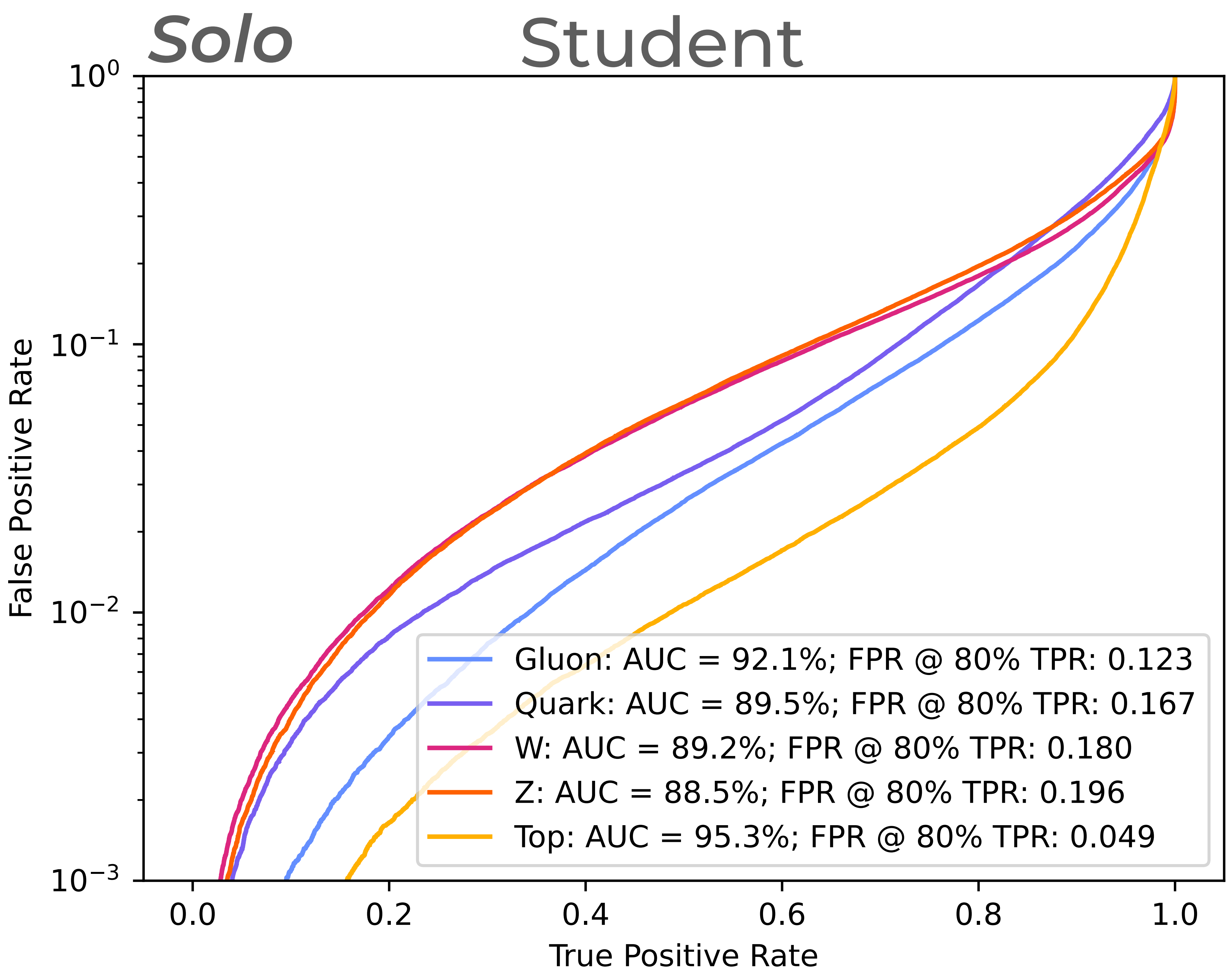


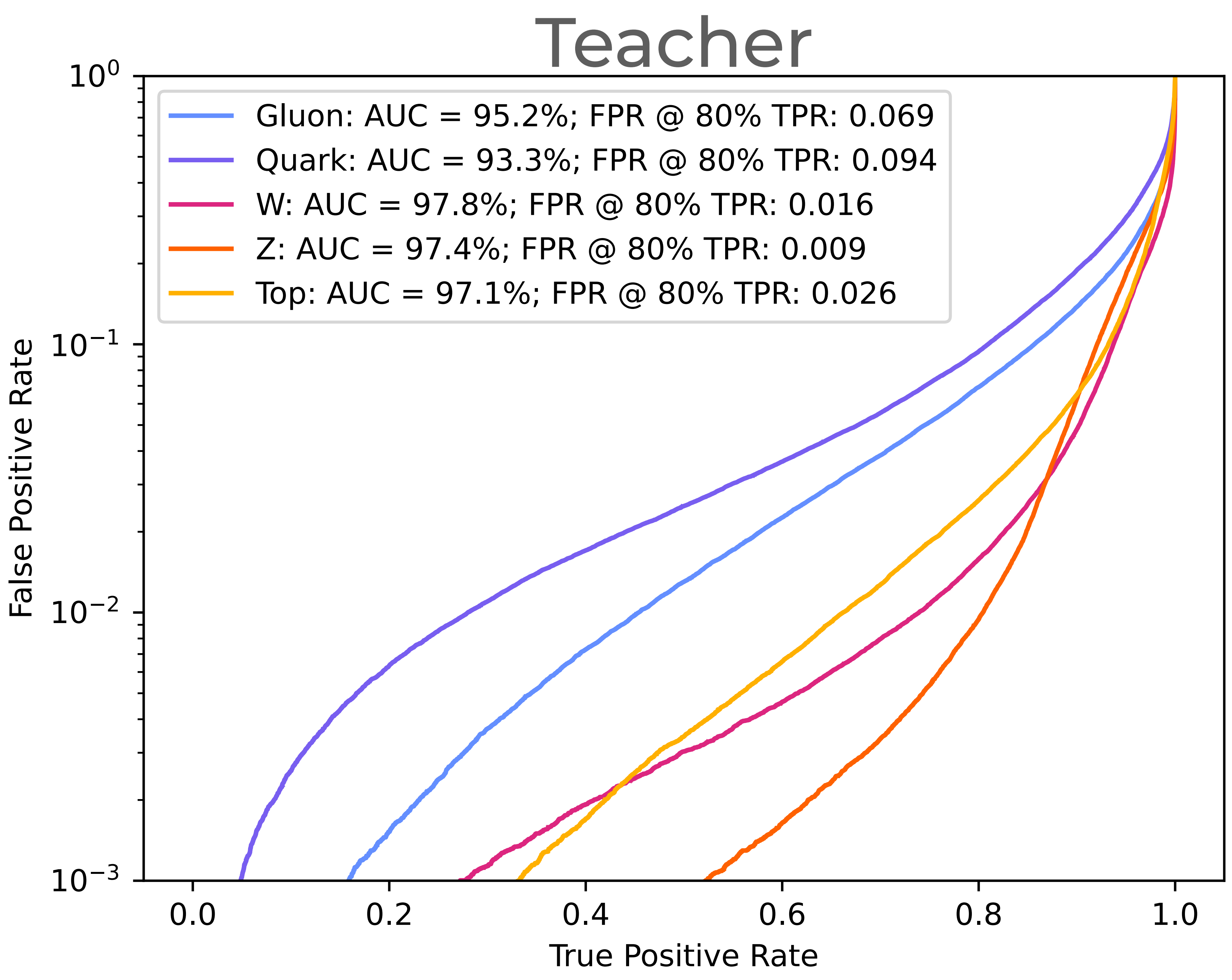

Median: 36.0
Number of Jets
10^4
10^3
10^2
10^1
10^0
0
20
40
60
80
100
120
140
Number of Constituents

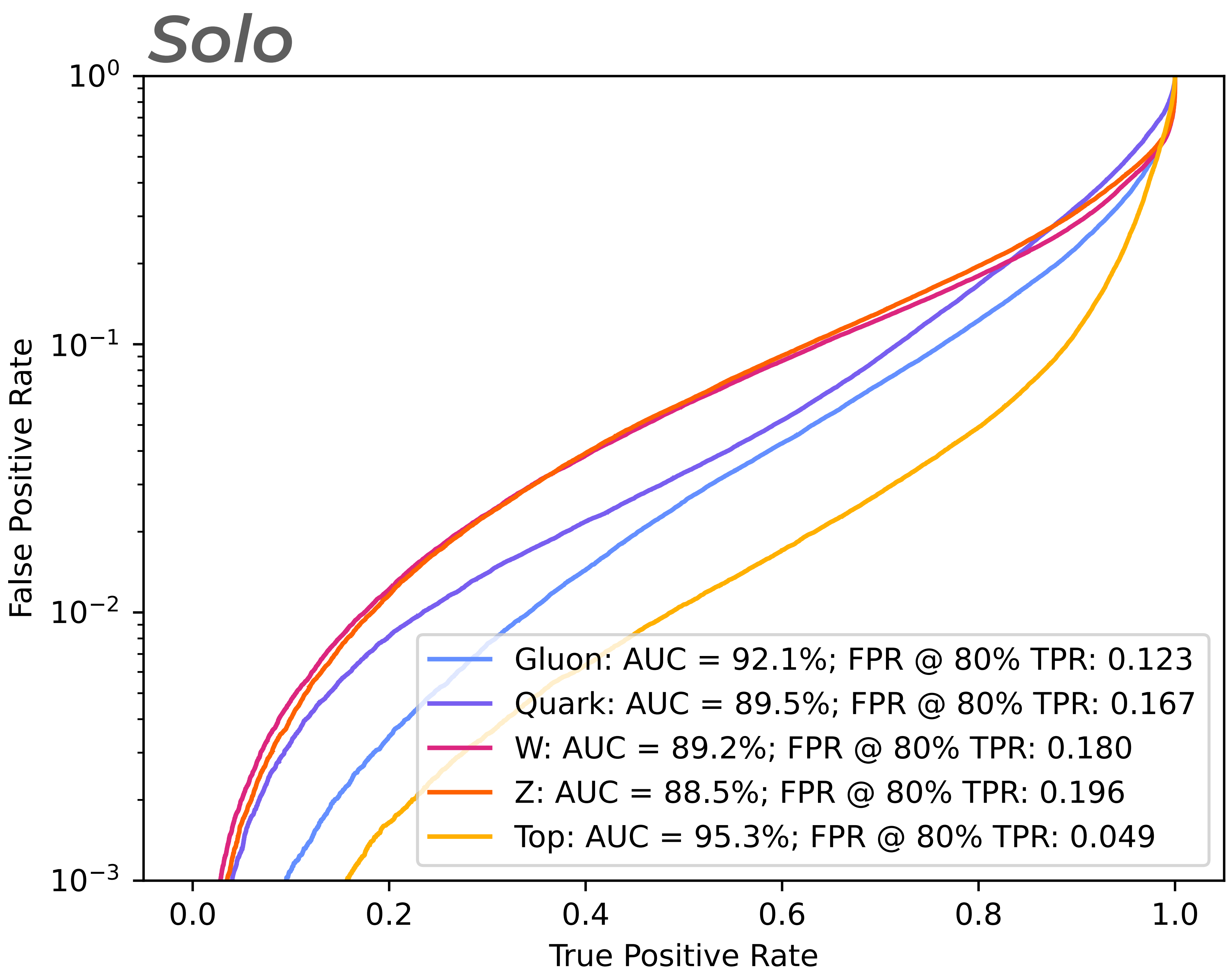
JEDInet DNN

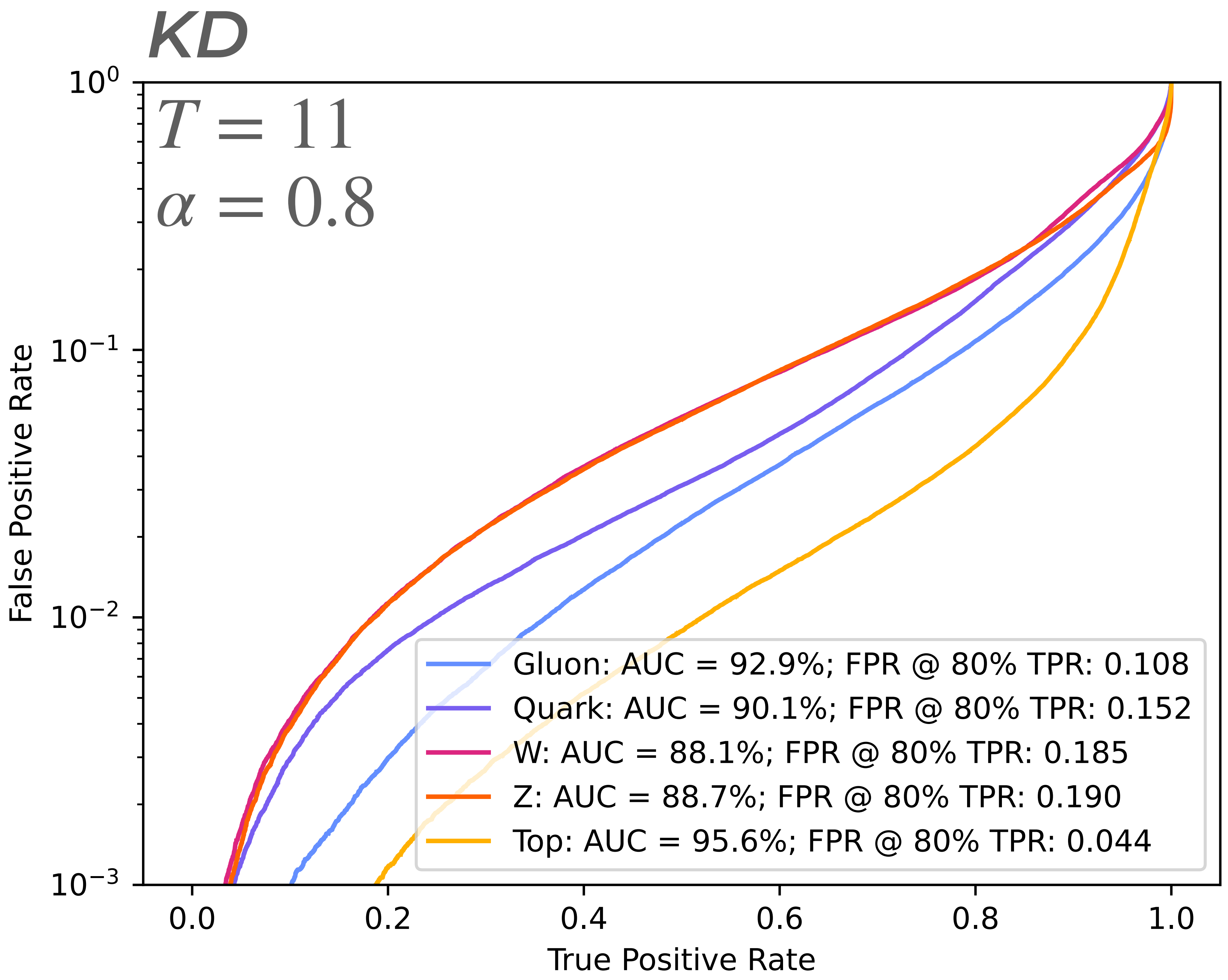
arxiv:1908.05318v3

50 constituents

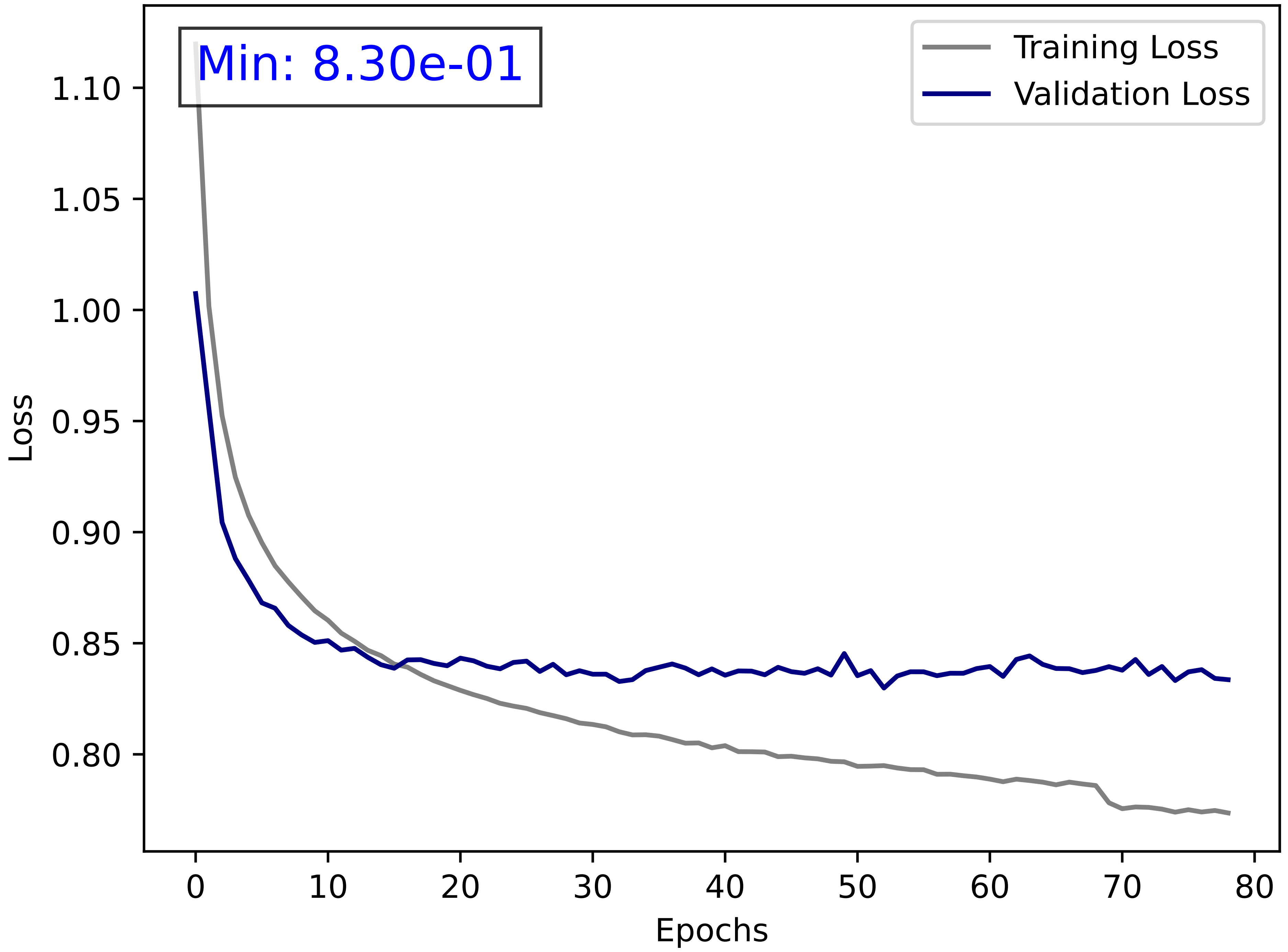





Solo
False Positive Rate
True Positive Rate
Gluon: AUC = 92.1%; FPR @ 80% TPR: 0.123
Quark: AUC = 89.5%; FPR @ 80% TPR: 0.167
W: AUC = 89.2%; FPR @ 80% TPR: 0.180
Z: AUC = 88.5%; FPR @ 80% TPR: 0.196
Top: AUC = 95.3%; FPR @ 80% TPR: 0.049

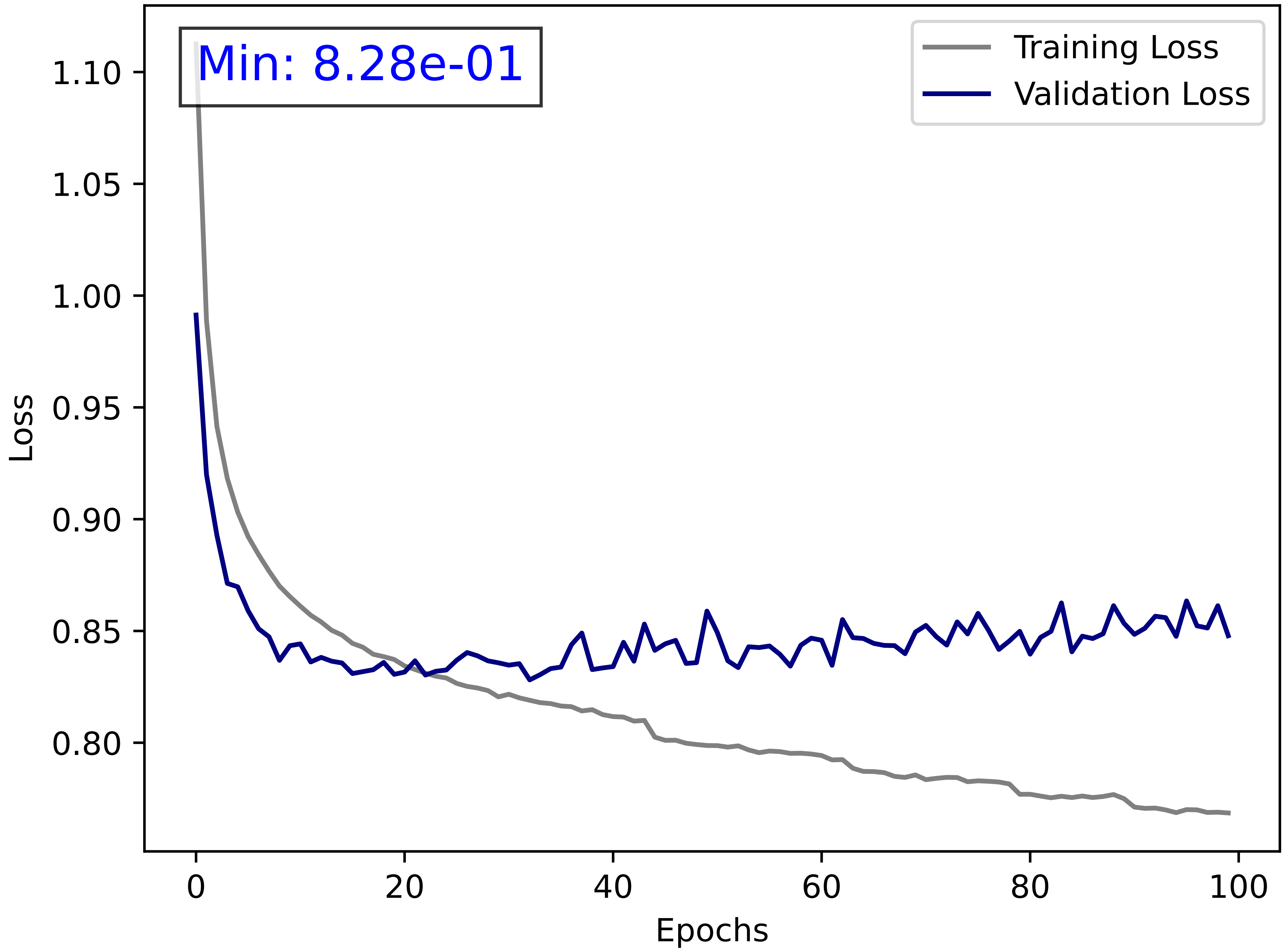

KD
$T = 11$
$\alpha = 0.8$
False Positive Rate
True Positive Rate
Gluon: AUC = 92.9%; FPR @ 80% TPR: 0.108
Quark: AUC = 90.1%; FPR @ 80% TPR: 0.152
W: AUC = 88.1%; FPR @ 80% TPR: 0.185
Z: AUC = 88.7%; FPR @ 80% TPR: 0.190
Top: AUC = 95.6%; FPR @ 80% TPR: 0.044

Solo



Min: 8.30e-01
Training Loss
Validation Loss
Loss
Epochs
1.10
1.05
1.00
0.95
0.90
0.85
0.80
0
10
20
30
40
50
60
70
80

KD

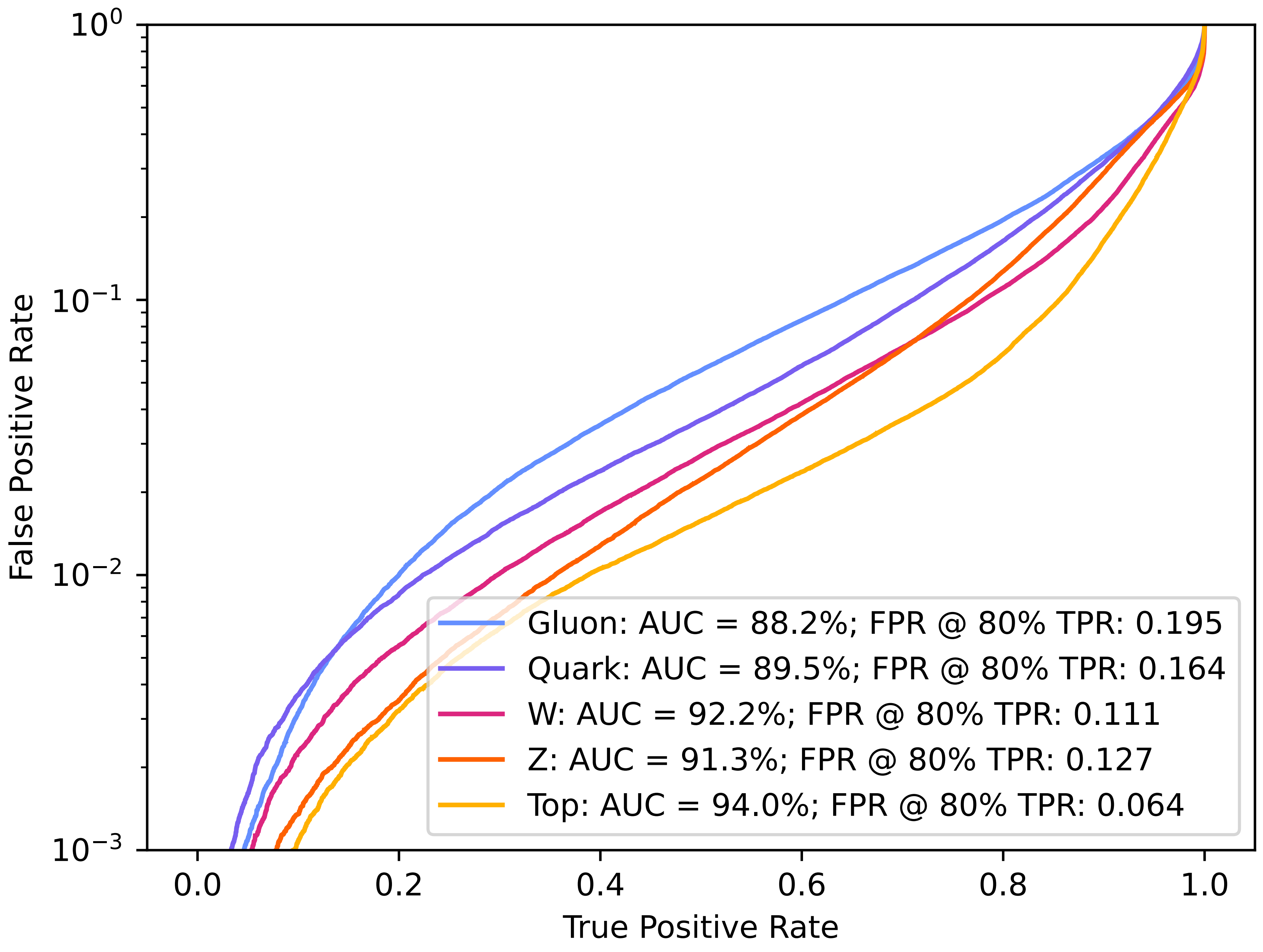


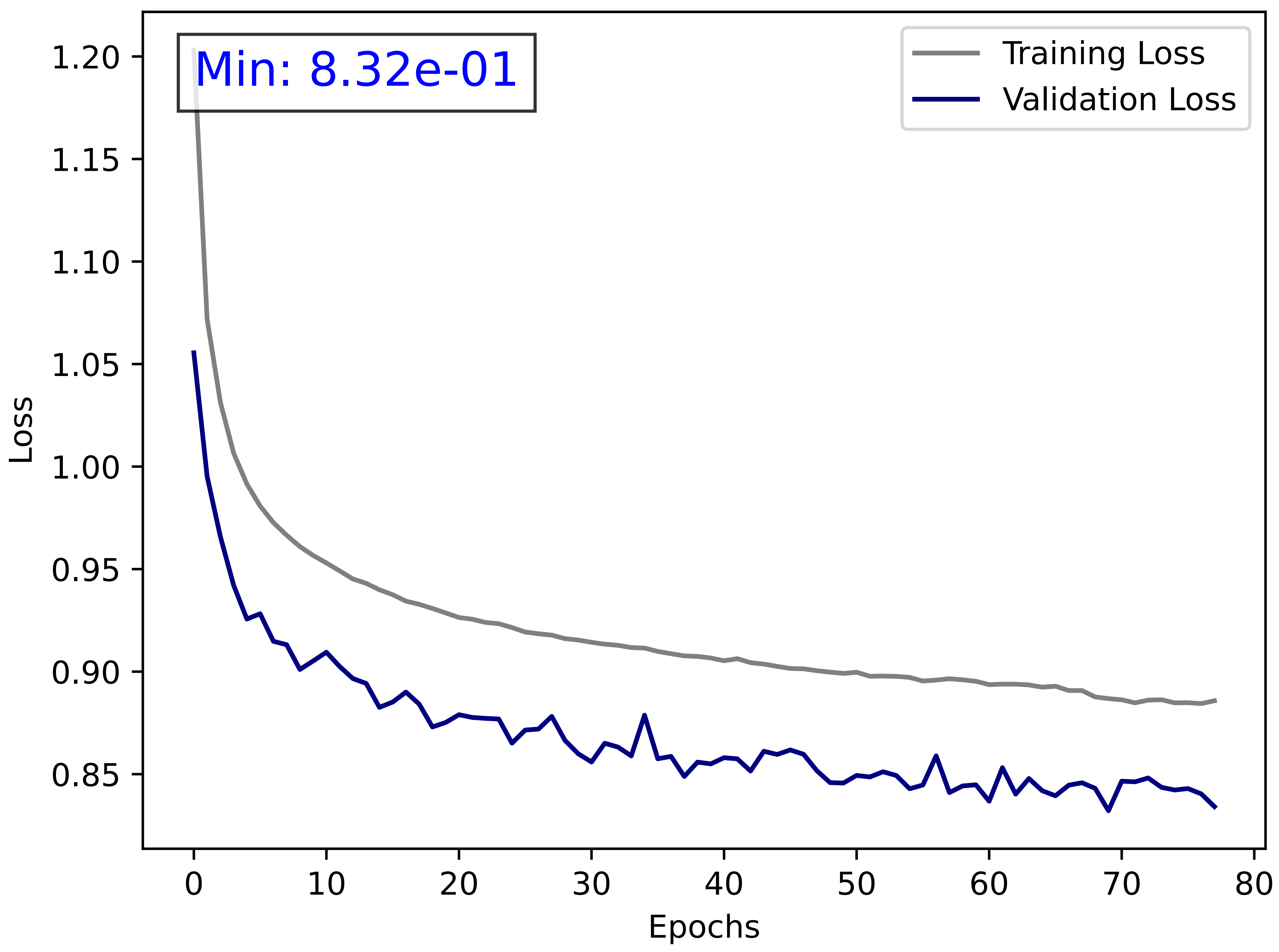
Min: 8.28e-01
Training Loss
Validation Loss
Loss
Epochs
1.10
1.05
1.00
0.95
0.90
0.85
0.80
0
20
40
60
80
100

JEDInet DNN

arxiv:1908.05318v3

16 constituents 3 features (pT, eta, phi) - pT ordered

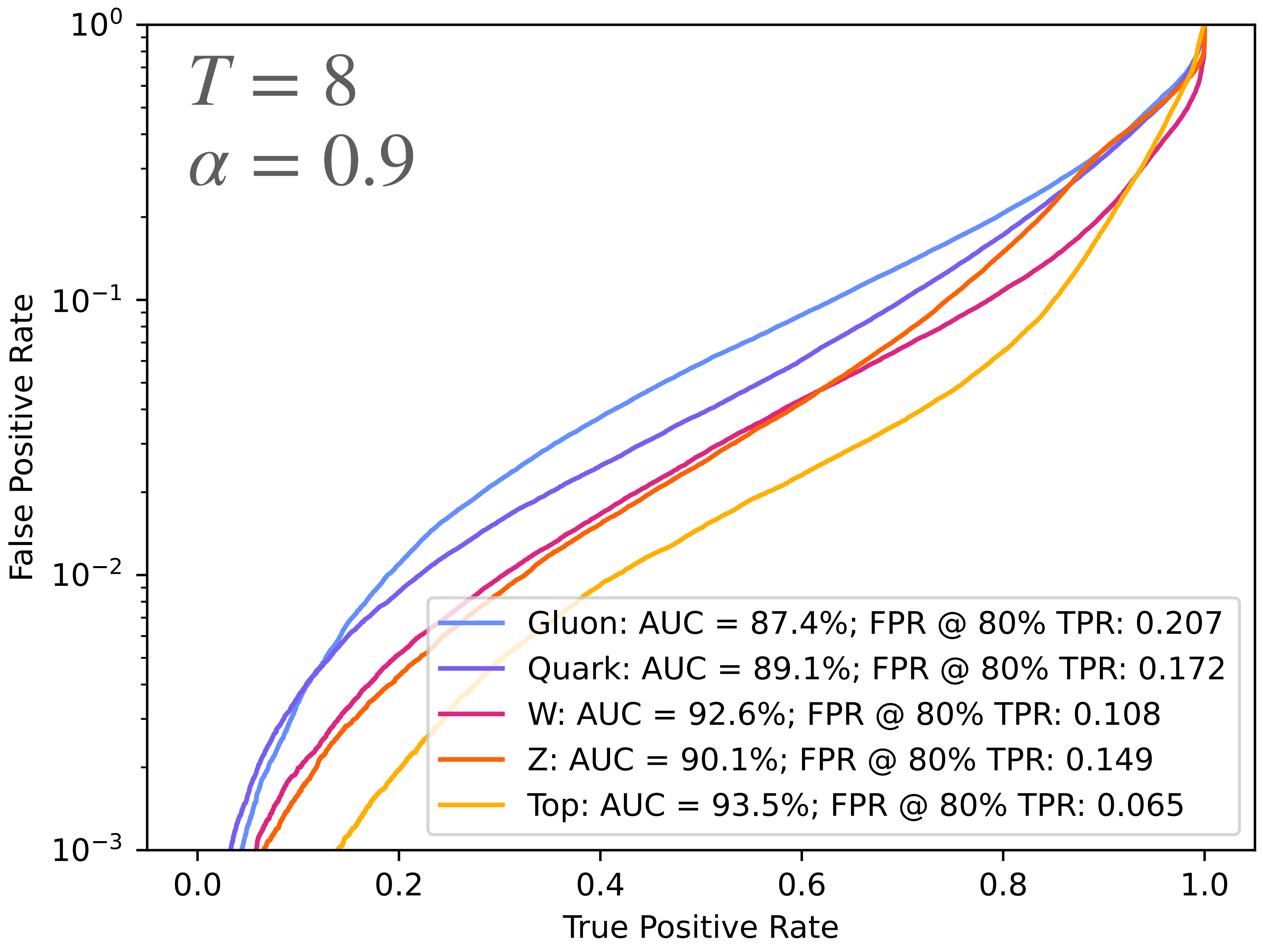


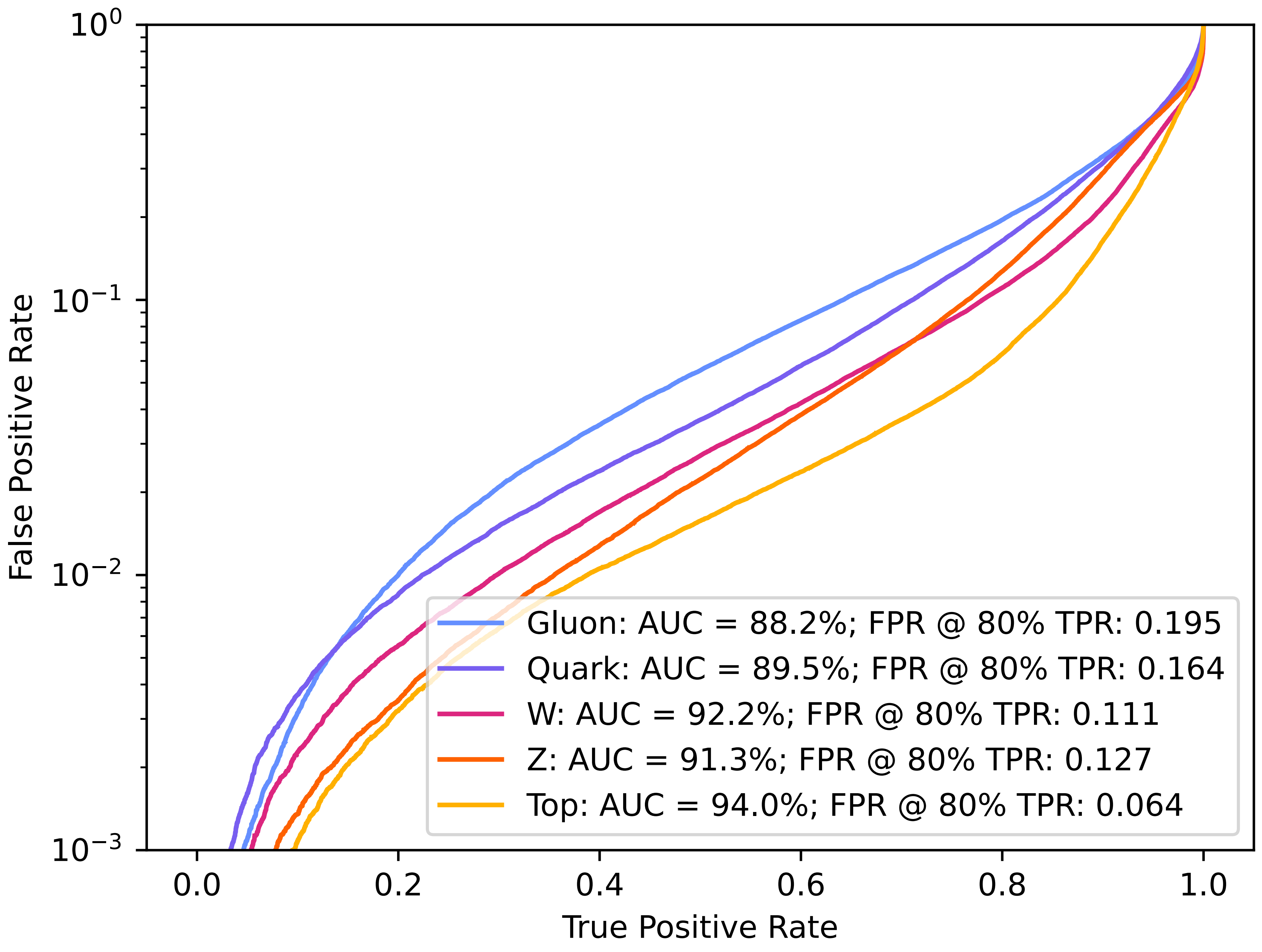


JEDInet DNN

arxiv:1908.05318v3

150 constituents 16 features IntNet teacher
16 constituents 3 feature (pt, eta, phi) - pT ordered

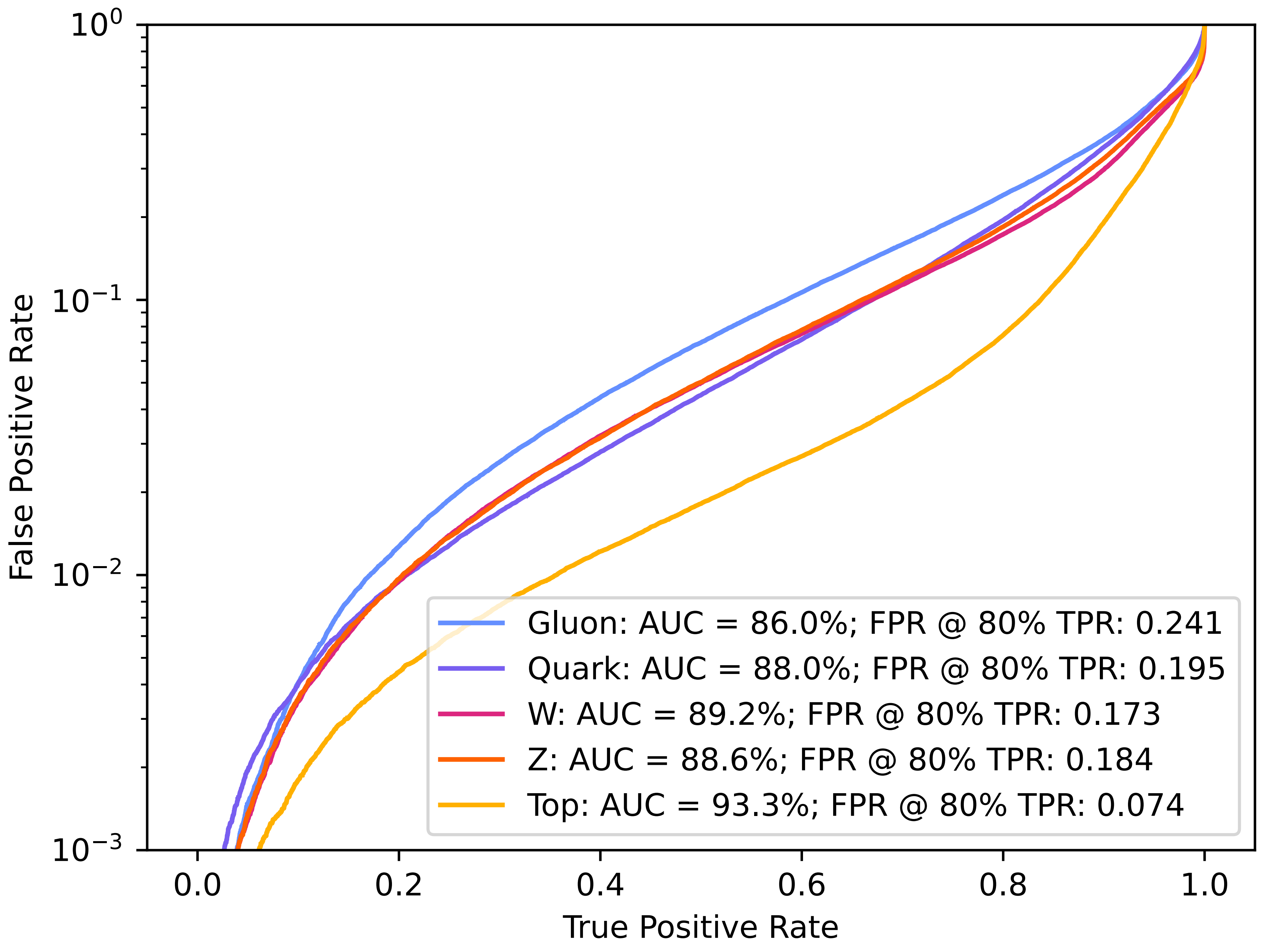


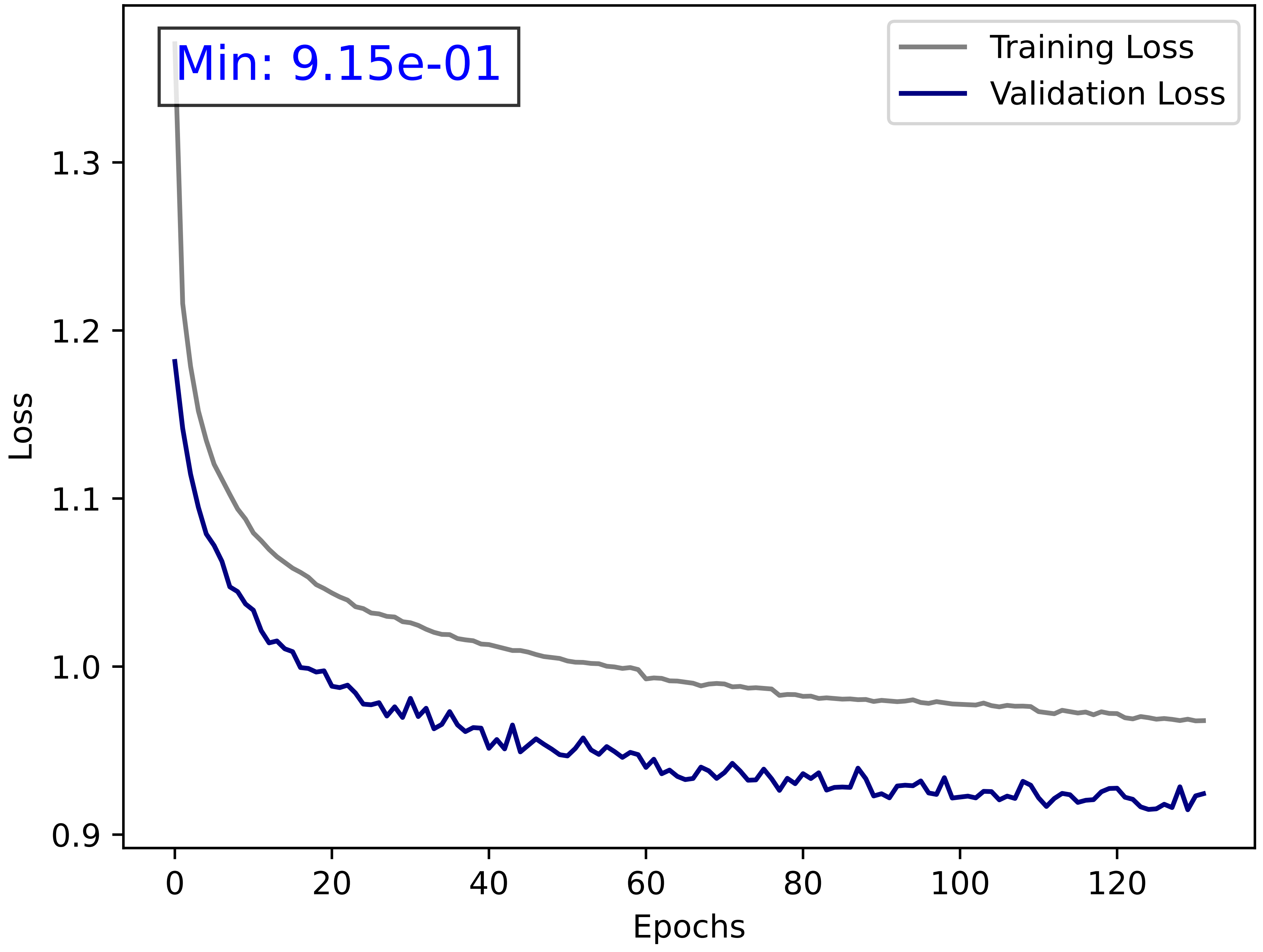


JEDInet DNN

arxiv:1908.05318v3

16 constituents 3 feature (pt, eta, phi) - shuffled

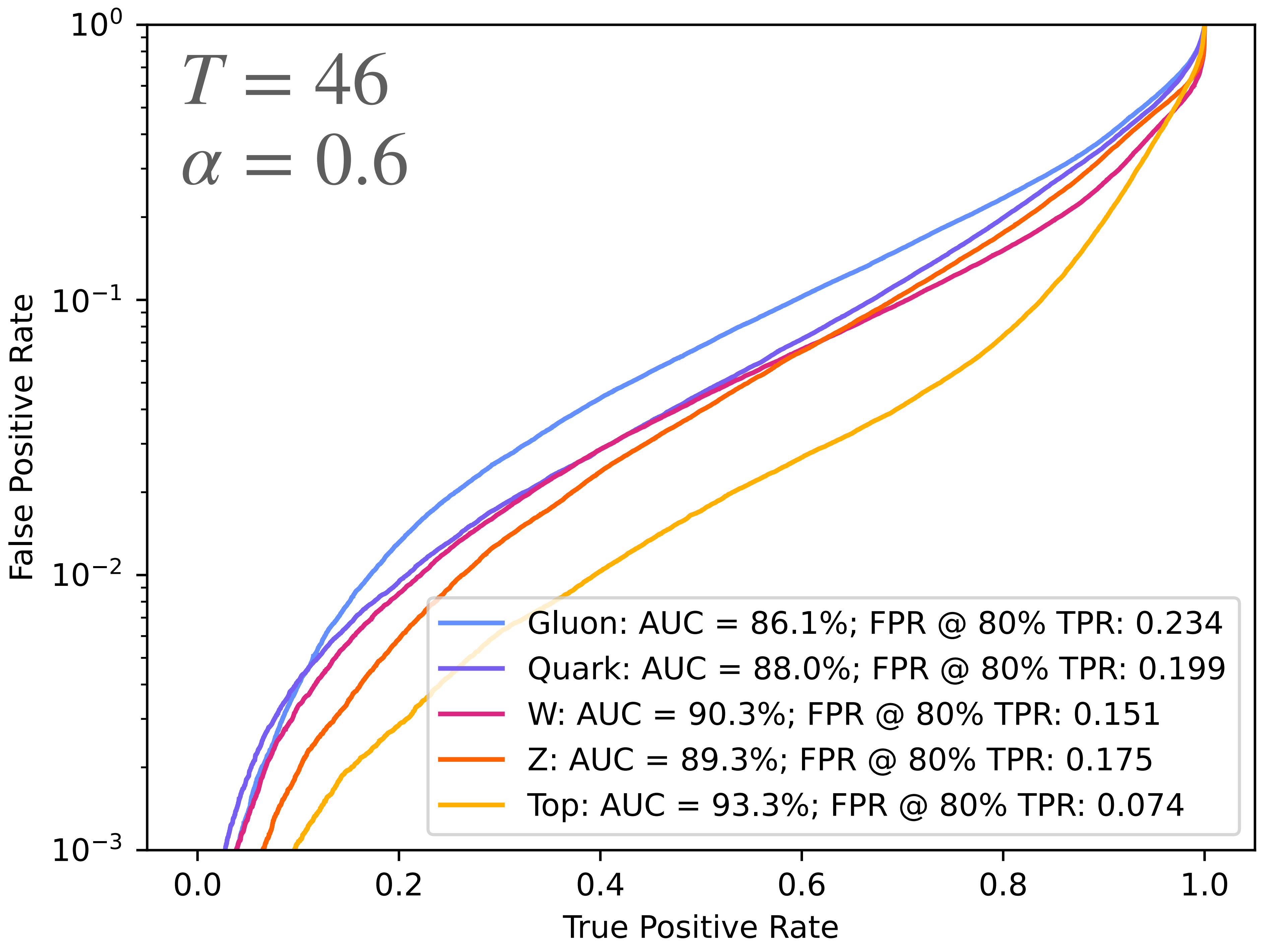


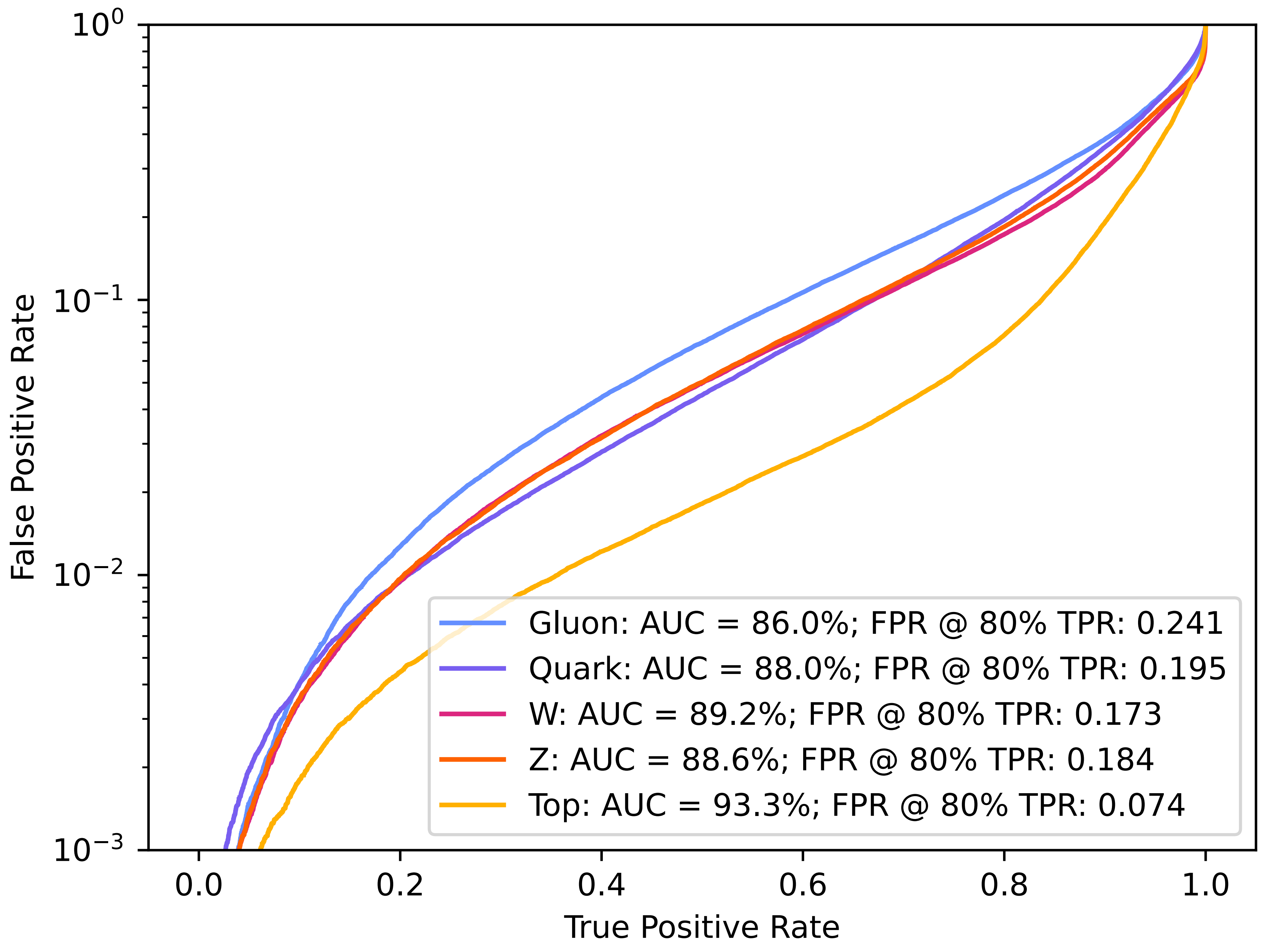


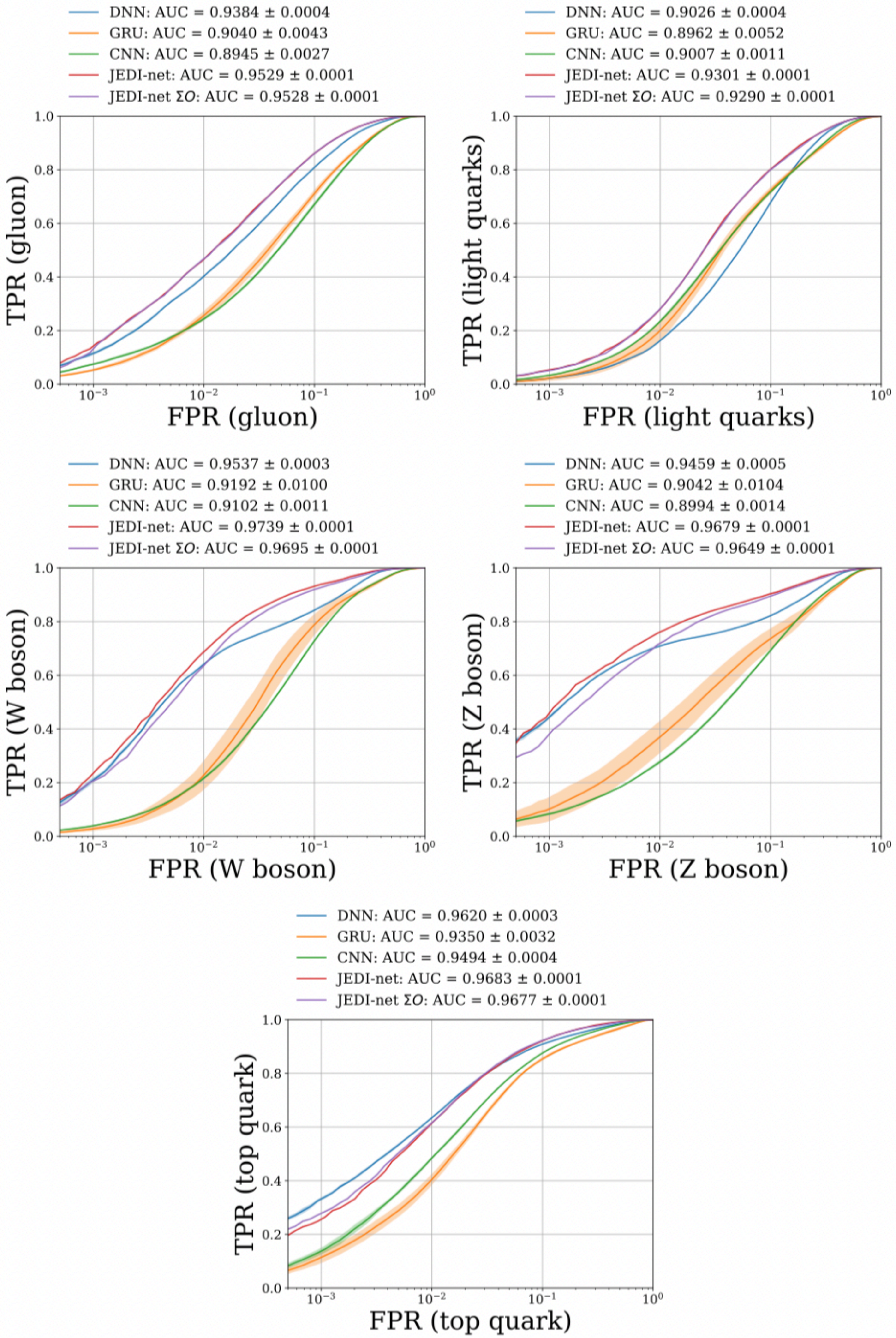
JEDInet DNN

arxiv:1908.05318v3

150 constituents 16 features int net teacher
16 constituents 3 feature (pt, eta, phi) - shuffled






DNN: AUC = 0.9384 ± 0.0004
GRU: AUC = 0.9040 ± 0.0043
CNN: AUC = 0.8945 ± 0.0027
JEDI-net: AUC = 0.9529 ± 0.0001
JEDI-net ΣO: AUC = 0.9528 ± 0.0001
TPR (gluon)
FPR (gluon)
DNN: AUC = 0.9026 ± 0.0004
GRU: AUC = 0.8962 ± 0.0052
CNN: AUC = 0.9007 ± 0.0011
JEDI-net: AUC = 0.9301 ± 0.0001
JEDI-net ΣO: AUC = 0.9290 ± 0.0001
TPR (light quarks)
FPR (light quarks)
DNN: AUC = 0.9537 ± 0.0003
GRU: AUC = 0.9192 ± 0.0100
CNN: AUC = 0.9102 ± 0.0011
JEDI-net: AUC = 0.9739 ± 0.0001
JEDI-net ΣO: AUC = 0.9695 ± 0.0001
TPR (W boson)
FPR (W boson)
DNN: AUC = 0.9459 ± 0.0005
GRU: AUC = 0.9042 ± 0.0104
CNN: AUC = 0.8994 ± 0.0014
JEDI-net: AUC = 0.9679 ± 0.0001
JEDI-net ΣO: AUC = 0.9649 ± 0.0001
TPR (Z boson)
FPR (Z boson)
DNN: AUC = 0.9620 ± 0.0003
GRU: AUC = 0.9350 ± 0.0032
CNN: AUC = 0.9494 ± 0.0004
JEDI-net: AUC = 0.9683 ± 0.0001
JEDI-net ΣO: AUC = 0.9677 ± 0.0001
TPR (top quark)
FPR (top quark)