The Good, the Bad, and the Bayesian

How networks learn uncertainties

Nina Elmer

ML4Jets 2024

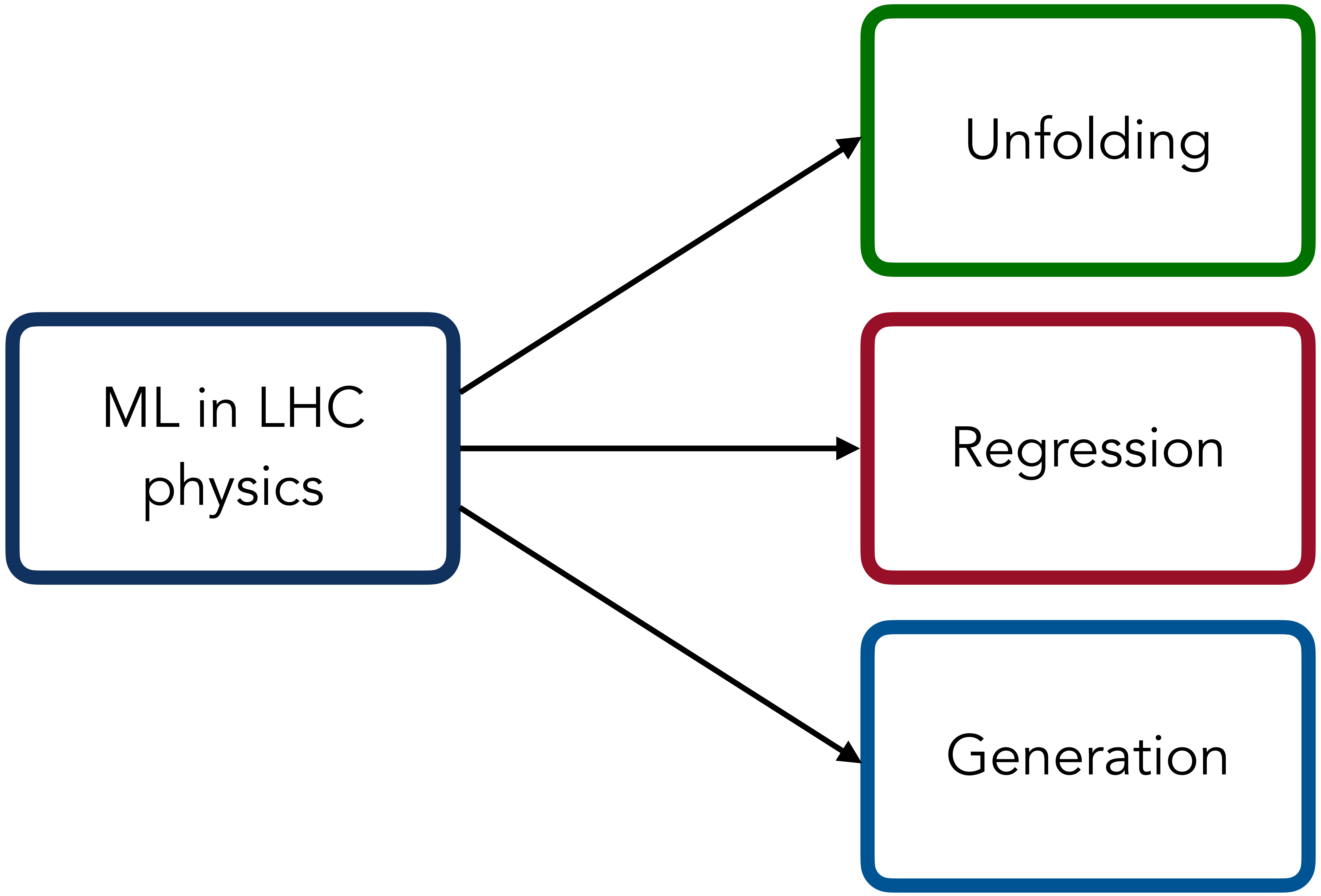
arXiv: 2412.xxxxx

with L. Favaro, M. Haußmann, R. Winterhalder and T. Plehn

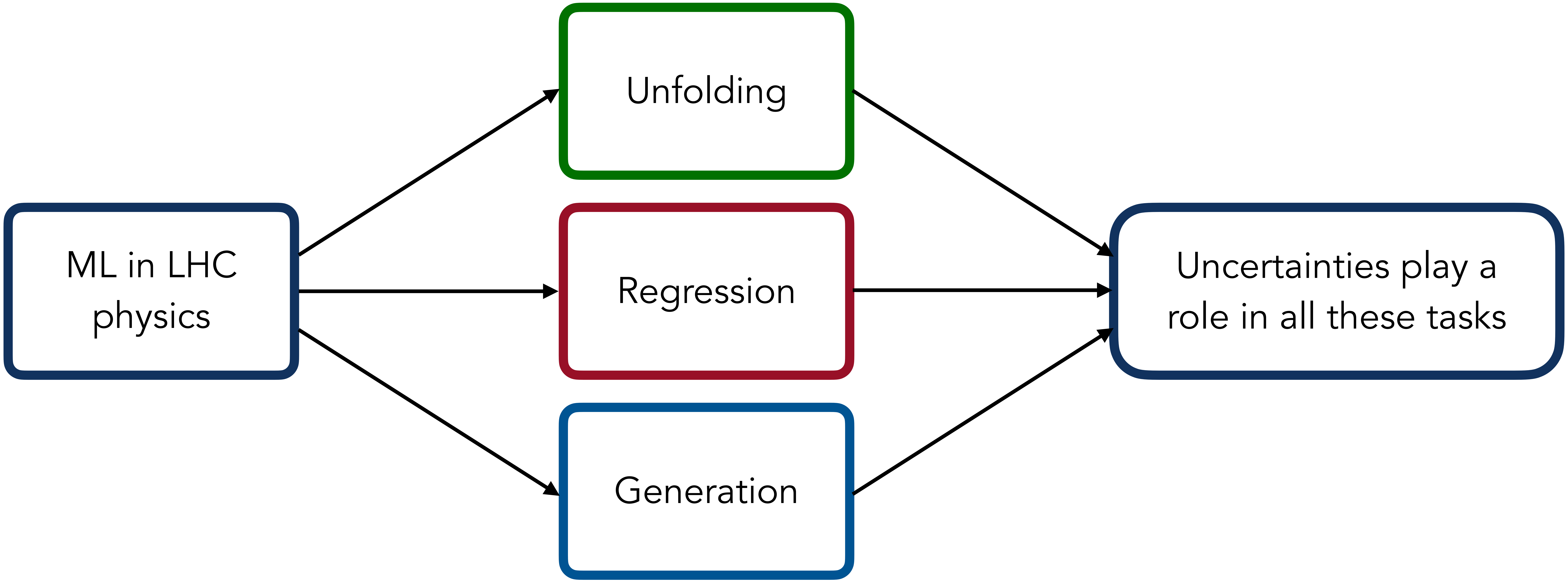
UNIVERSITÄT HEIDELBERG
ZUKUNFT
SEIT 1386

IMPRS
for Precision Tests of Fundamental Symmetries
INTERNATIONAL MAX PLANCK RESEARCH SCHOOL

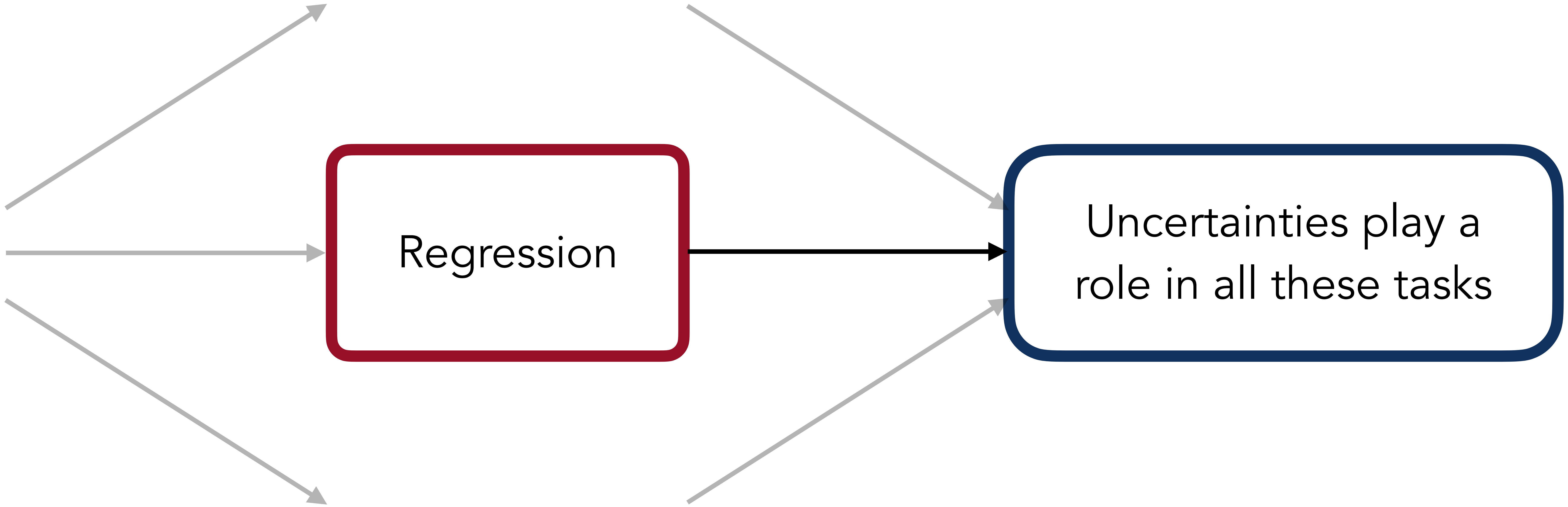
Motivation



Motivation


ML in LHC physics
Unfolding
Regression
Generation
Uncertainties play a role in all these tasks

Motivation



Motivation

- Fit set of Amplitudes $A(x)$, with training data: $\{x, A(x)\}$

Motivation

- Fit set of Amplitudes $A(x)$, with training data: $\{x, A(x)\}$
- Prediction in regression:

Motivation

- Fit set of Amplitudes $A(x)$, with training data: $\{x, A(x)\}$
- Prediction in regression:

$$A(x) \equiv \langle A \rangle = \int \mathrm{d}A\, A\, p(A\,|\,x) = \int \mathrm{d}\theta\, q(\theta)\, \boxed{\bar{A}(x,\theta)} \quad \text{with} \quad p(A\,|\,x) = \int \mathrm{d}\theta\, p(A\,|\,\theta, x)\, p(\theta\,|\,x)$$

network output

Motivation

- Fit set of Amplitudes $A(x)$, with training data: $\{x, A(x)\}$
- Prediction in regression:

$$A(x) \equiv \langle A \rangle = \int \mathrm{d}A\, A\, p(A\,|\,x) = \int \mathrm{d}\theta\, q(\theta)\, \boxed{\overline{A}(x,\theta)} \qquad \text{with} \qquad p(A\,|\,x) = \int \mathrm{d}\theta\, p(A\,|\,\theta, x)\, p(\theta\,|\,x)$$

network output

$$\sigma_{\mathrm{tot}}^2(x) \equiv \langle (A - \langle A \rangle)^2 \rangle = \int \mathrm{d}A\, \left(A - \langle A \rangle\right)^2 p(A\,|\,x) = \int \mathrm{d}\theta\, q(\theta) \left(\overline{A^2}(x,\theta) - \overline{A}(x,\theta)^2 \right) + \int \mathrm{d}\theta\, q(\theta) \left(\overline{A}(x,\theta) - \langle A \rangle \right)^2$$

Motivation

- Fit set of Amplitudes $A(x)$, with training data: $\{x, A(x)\}$
- Prediction in regression:

$$A(x) \equiv \langle A \rangle = \int dA\, A\, p(A\,|\,x) = \int d\theta\, q(\theta)\, \boxed{\overline{A}(x,\theta)} \quad \text{with} \quad p(A\,|\,x) = \int d\theta\, p(A\,|\,\theta, x)\, p(\theta\,|\,x)$$

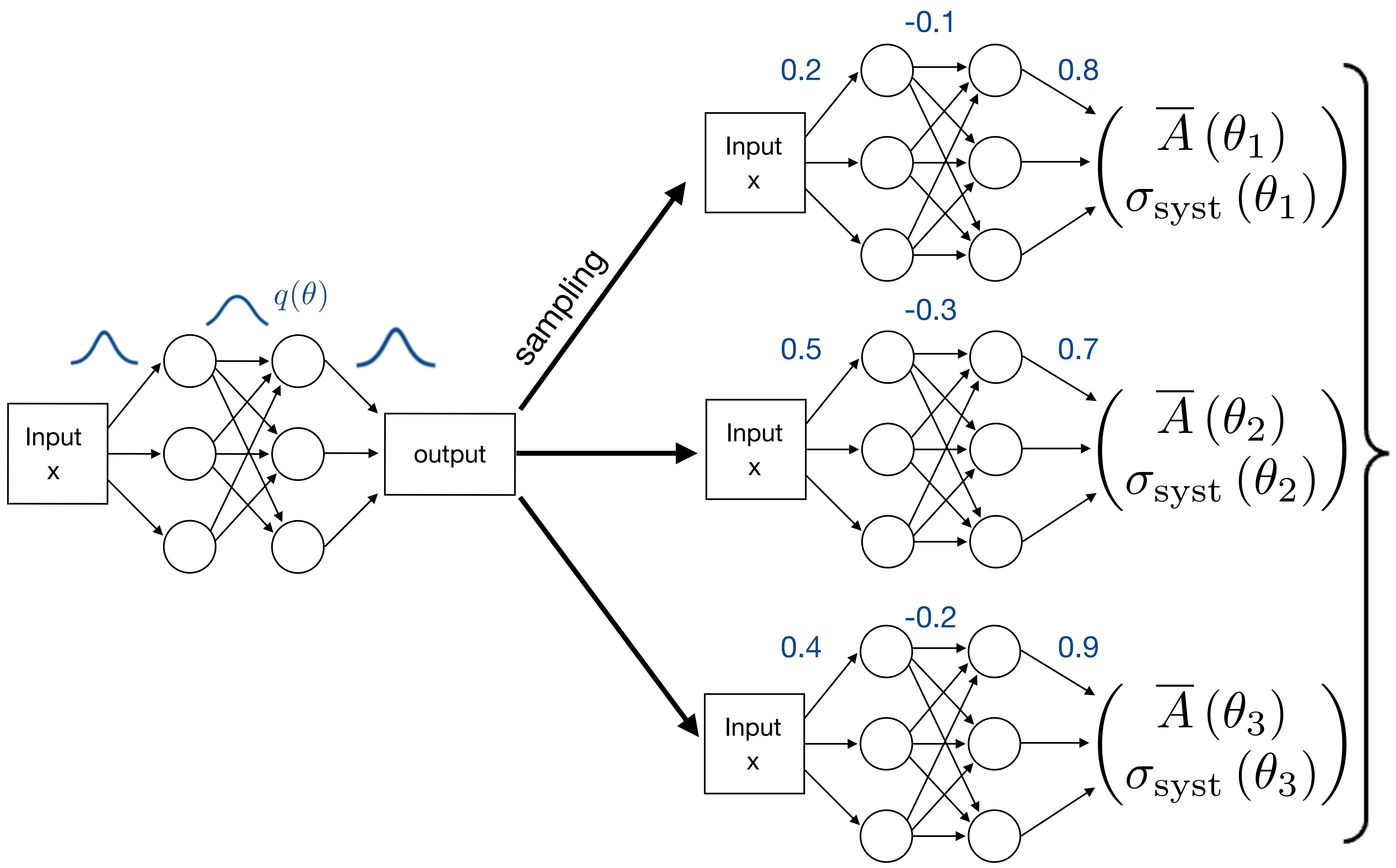
network output

$$\sigma^2_{\text{tot}}(x) \equiv \langle (A - \langle A \rangle)^2 \rangle = \int dA\, \left(A - \langle A \rangle\right)^2 p(A\,|\,x) = \int d\theta\, q(\theta) \boxed{\left(\overline{A^2}(x,\theta) - \overline{A}(x,\theta)^2\right)} + \int d\theta\, q(\theta) \left(\overline{A}(x,\theta) - \langle A \rangle\right)^2$$

Gaussian uncertainty in heteroscedastic loss: $\mathscr{L}_{\text{heteroscedastic}} = \sum_i \frac{|f(x_i) - f_\theta(x_i)|^2}{2\sigma(x_i)^2} + \log \sigma(x_i) + \ldots$

Bayesian neural network (BNN)

BNN **Ensemble of networks** **Output**



$$\langle A \rangle = \frac{1}{N} \sum_i^N \overline{A}(\theta_i)$$

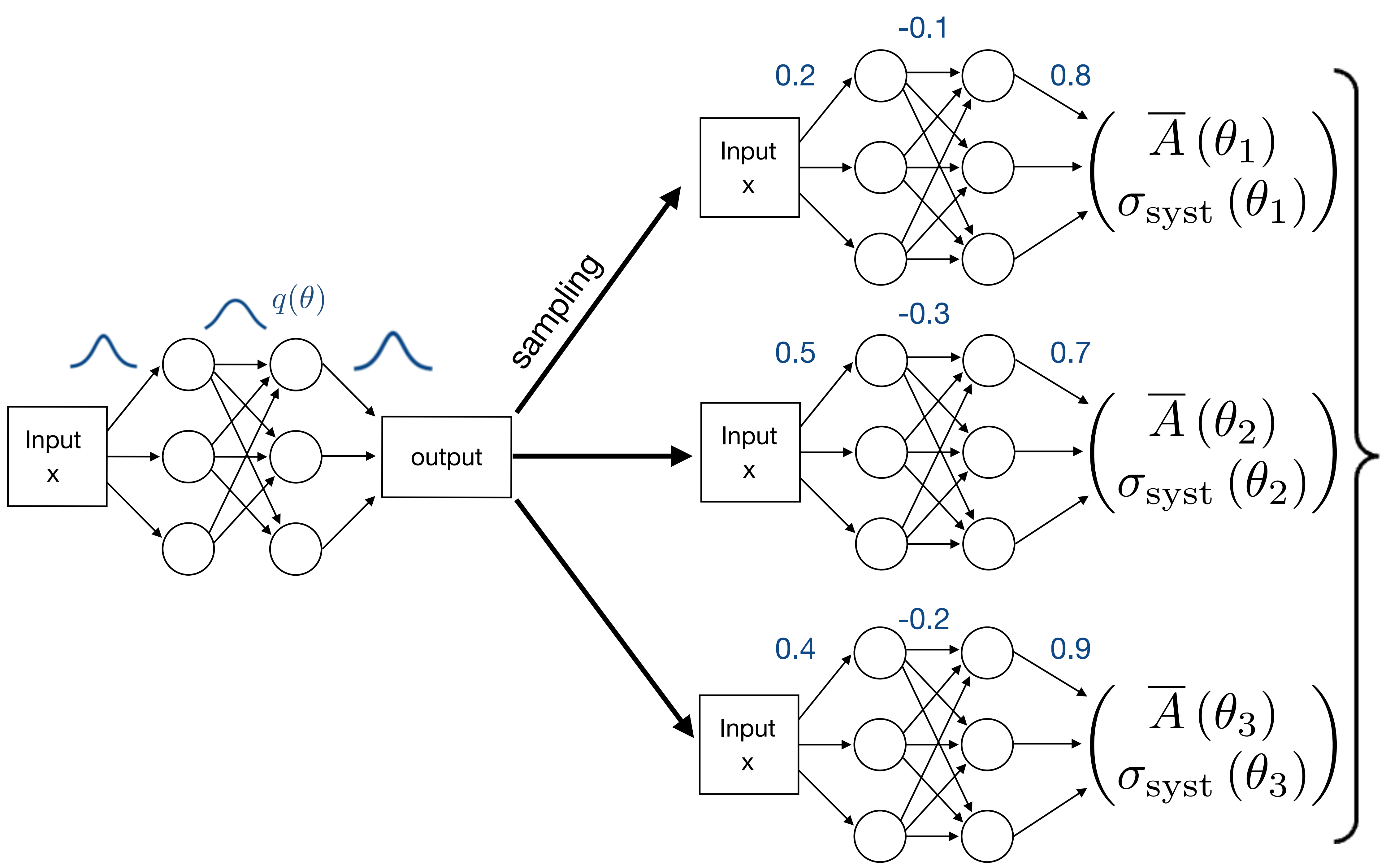
$$\sigma_{\text{syst}}^2 = \frac{1}{N} \sum_i^N \sigma_{\text{syst}}^2(\theta_i)$$

$$\sigma_{\text{stat}}^2 = \frac{1}{N} \sum_i^N \left(\langle A \rangle - \overline{A}(\theta_i)\right)^2$$

$$\mathcal{L}_{\text{BNN}} = \sum_x \Big[\text{KL}[q(\theta), p(\theta)] - \big\langle \log p(D_{\text{train}}|\theta) \big\rangle_{\theta \sim q(\theta)} \Big]$$

Bayesian neural network (BNN)

BNN **Ensemble of networks** **Output**



$$\langle A \rangle = \frac{1}{N} \sum_i^N \bar{A}(\theta_i)$$

$$\sigma^2_{\text{syst}} = \frac{1}{N} \sum_i^N \sigma^2_{\text{syst}}(\theta_i)$$

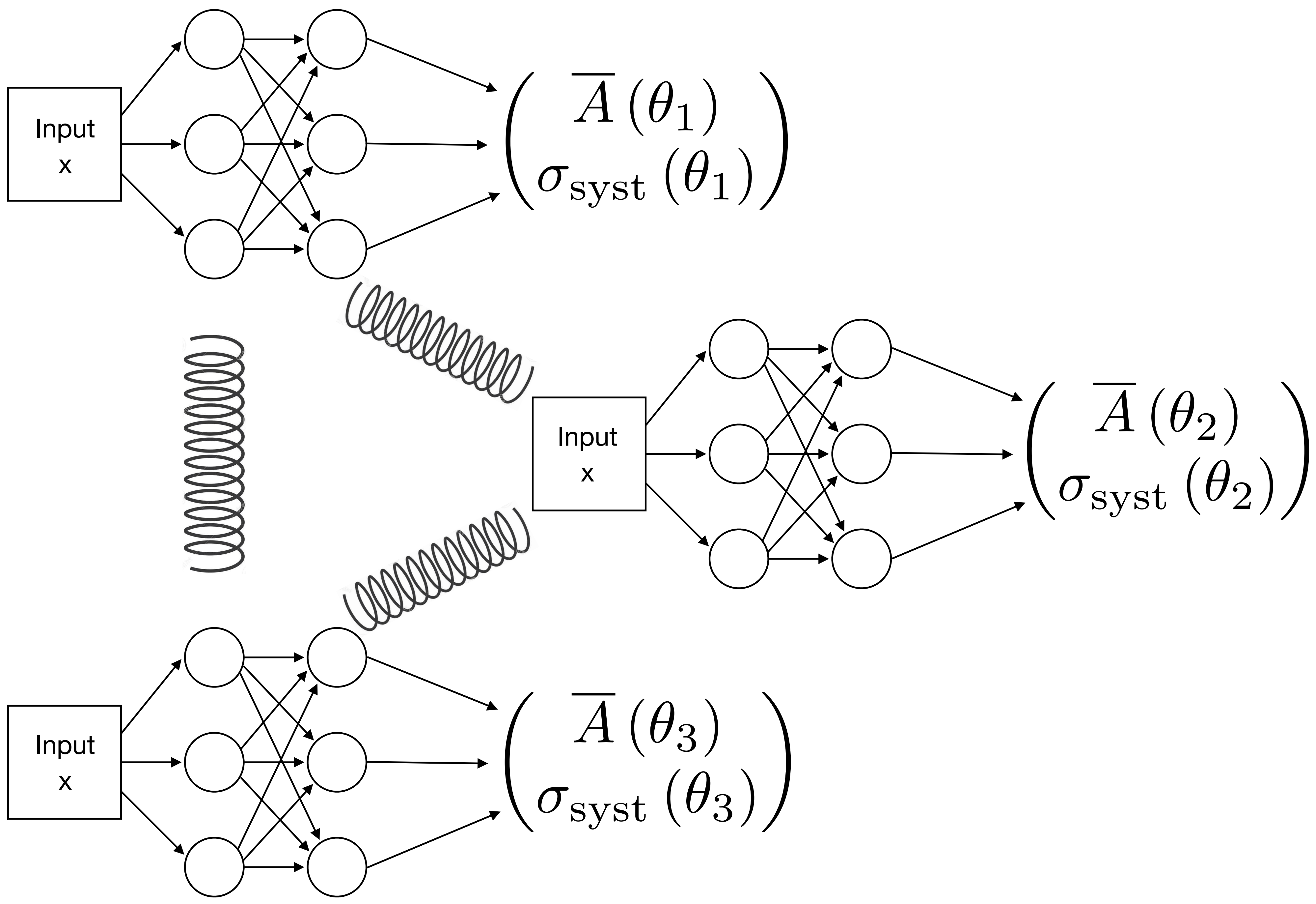
$$\sigma^2_{\text{stat}} = \frac{1}{N} \sum_i^N \left(\langle A \rangle - \bar{A}(\theta_i)\right)^2$$

$$\mathcal{L}_{\text{BNN}} = \sum_x \Big[\text{KL}[q(\theta), p(\theta)] - \big\langle \log p(D_{\text{train}}|\theta) \big\rangle_{\theta \sim q(\theta)} \Big]$$

- Parameters: **Network weights $q(\theta)$**
- $q(\theta)$ params of a Gaussian distribution
- Ensemble: Sample from weight distribution

Repulsive ensemble (RE)

Ensemble of networks **Output**



$$\langle A \rangle = \frac{1}{N} \sum_i^N \overline{A}(\theta_i)$$

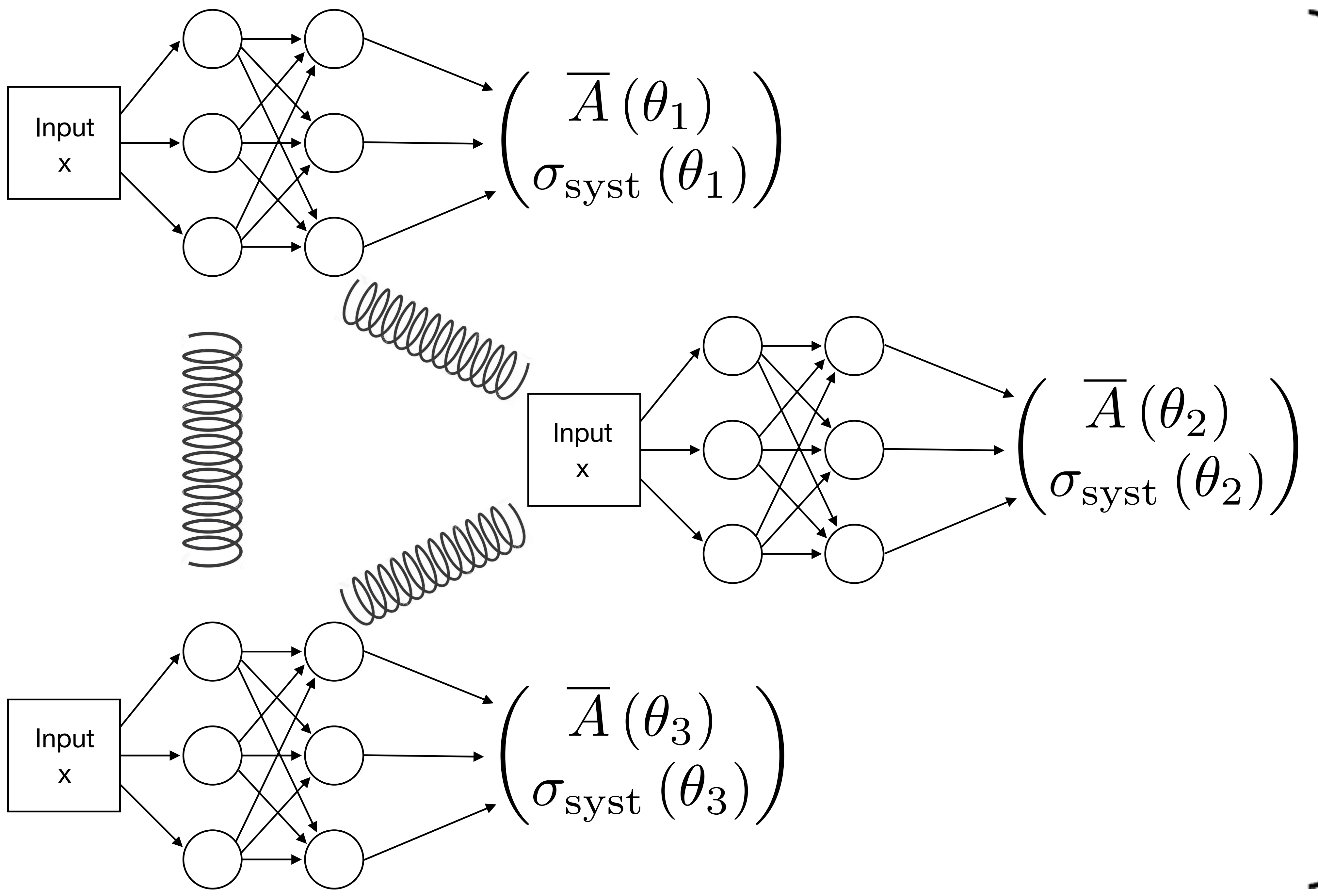
$$\sigma_{\text{syst}}^2 = \frac{1}{N} \sum_i^N \sigma_{\text{syst}}^2(\theta_i)$$

$$\sigma_{\text{stat}}^2 = \frac{1}{N} \sum_i^N \left(\langle A \rangle - \overline{A}(\theta_i) \right)^2$$

$$\mathcal{L}_{\text{RE}} = \sum_{i=1}^{n} \left[-\frac{1}{B} \sum_{b=1}^{B} \log p(x_b|\theta_i) + \frac{\beta}{N} \frac{\sum_{j=1}^{n} k(A_{\theta_i}(x), \overline{A_{\theta_j}(x)})}{\sum_{j=1}^{n} k(\overline{A_{\theta_i}(x)}, A_{\theta_j}(x))} + \frac{\theta_i^2}{2N\sigma^2} \right]$$

Repulsive ensemble (RE)

Ensemble of networks **Output**



$$\langle A \rangle = \frac{1}{N} \sum_i^N \overline{A}(\theta_i)$$

$$\sigma^2_{\text{syst}} = \frac{1}{N} \sum_i^N \sigma^2_{\text{syst}}(\theta_i)$$

$$\sigma^2_{\text{stat}} = \frac{1}{N} \sum_i^N \left(\langle A \rangle - \overline{A}(\theta_i) \right)^2$$

- Repulsive term: Cover full posterior distribution
- Ensemble members **trained simultaneously**

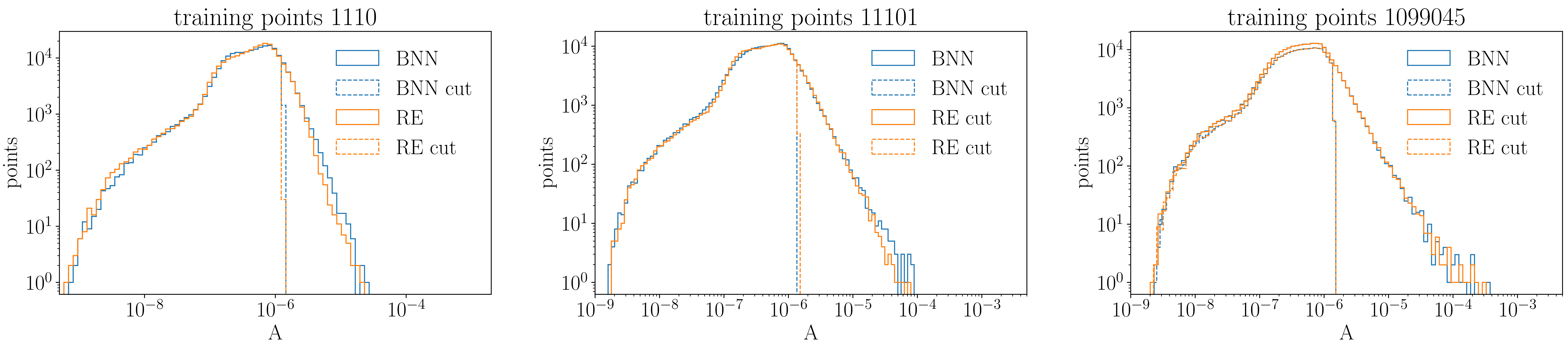
$$\mathcal{L}_{\text{RE}} = \sum_{i=1}^{n} \left[-\frac{1}{B} \sum_{b=1}^{B} \log p(x_b|\theta_i) + \frac{\beta}{N} \frac{\sum_{j=1}^{n} k(A_{\theta_i}(x), \overline{A_{\theta_j}(x)})}{\sum_{j=1}^{n} k(\overline{A_{\theta_i}(x)}, \overline{A_{\theta_j}(x)})} + \frac{\theta_i^2}{2N\sigma^2} \right]$$

Learning amplitudes

- Learning only Amplitudes for different training sizes
- Compare full data to all but 10% largest

Learning amplitudes

- Learning only Amplitudes for different training sizes
- Compare full data to all but 10% largest



Uncertainties

Two uncertainty types:

Systematic and **statistical**

Plateaus for perfect training

Vanishes with perfect training

Uncertainties

Two uncertainty types:

Systematic and **statistical**

Plateaus for perfect training

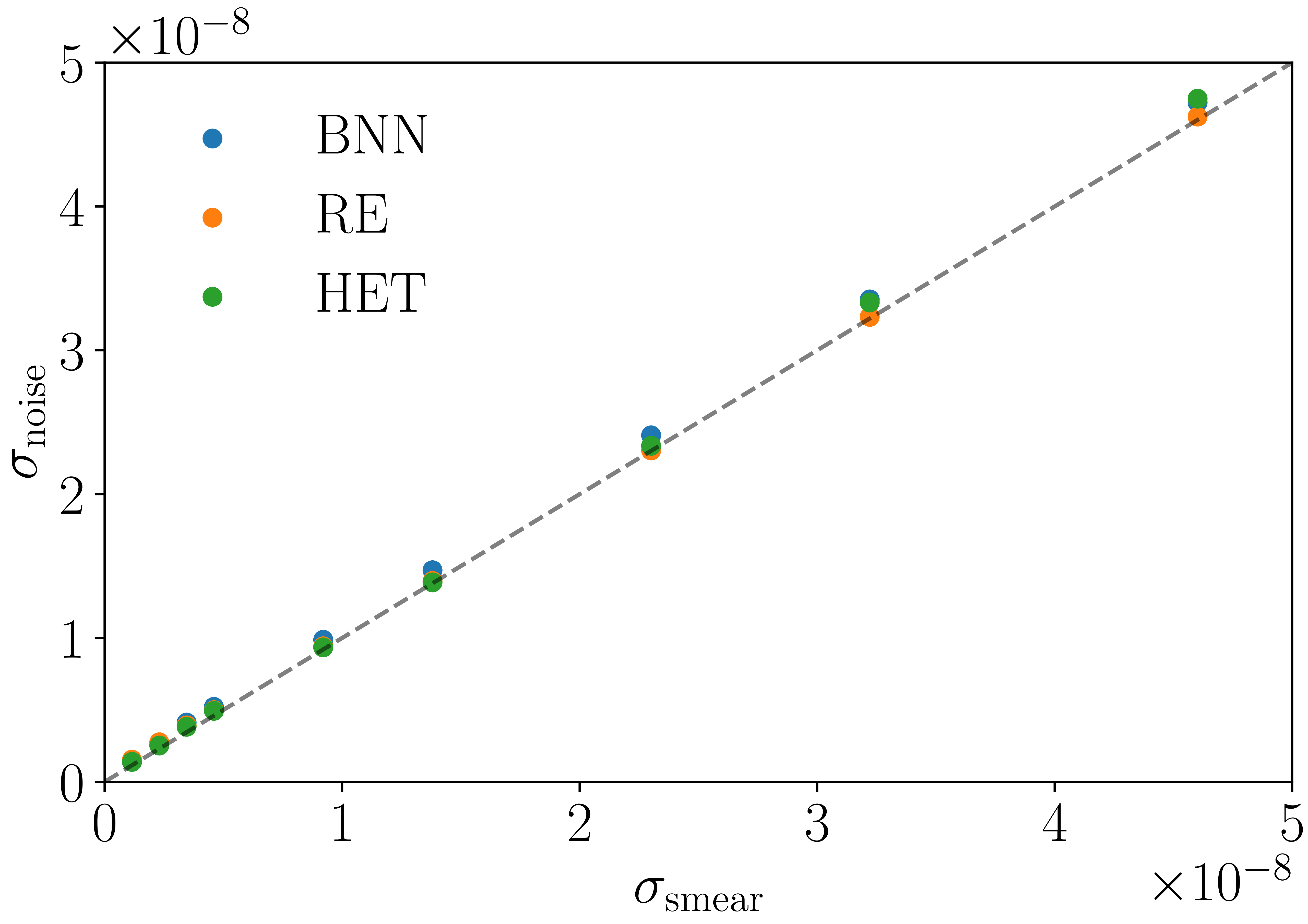
Vanishes with perfect training

$$\sigma^2_{\text{tot}} \equiv \sigma^2_{\text{syst}} + \sigma^2_{\text{stat}}$$

Adding Gaussian noise

$$\sigma^2_{\text{tot}} = \sigma^2_{\text{syst},0} + \sigma^2_{\text{noise}} + \sigma^2_{\text{stat}}$$

$$\sigma_{\text{smear}} = \{0.25, ..., 10\}\% \times A_{\text{truth}}$$



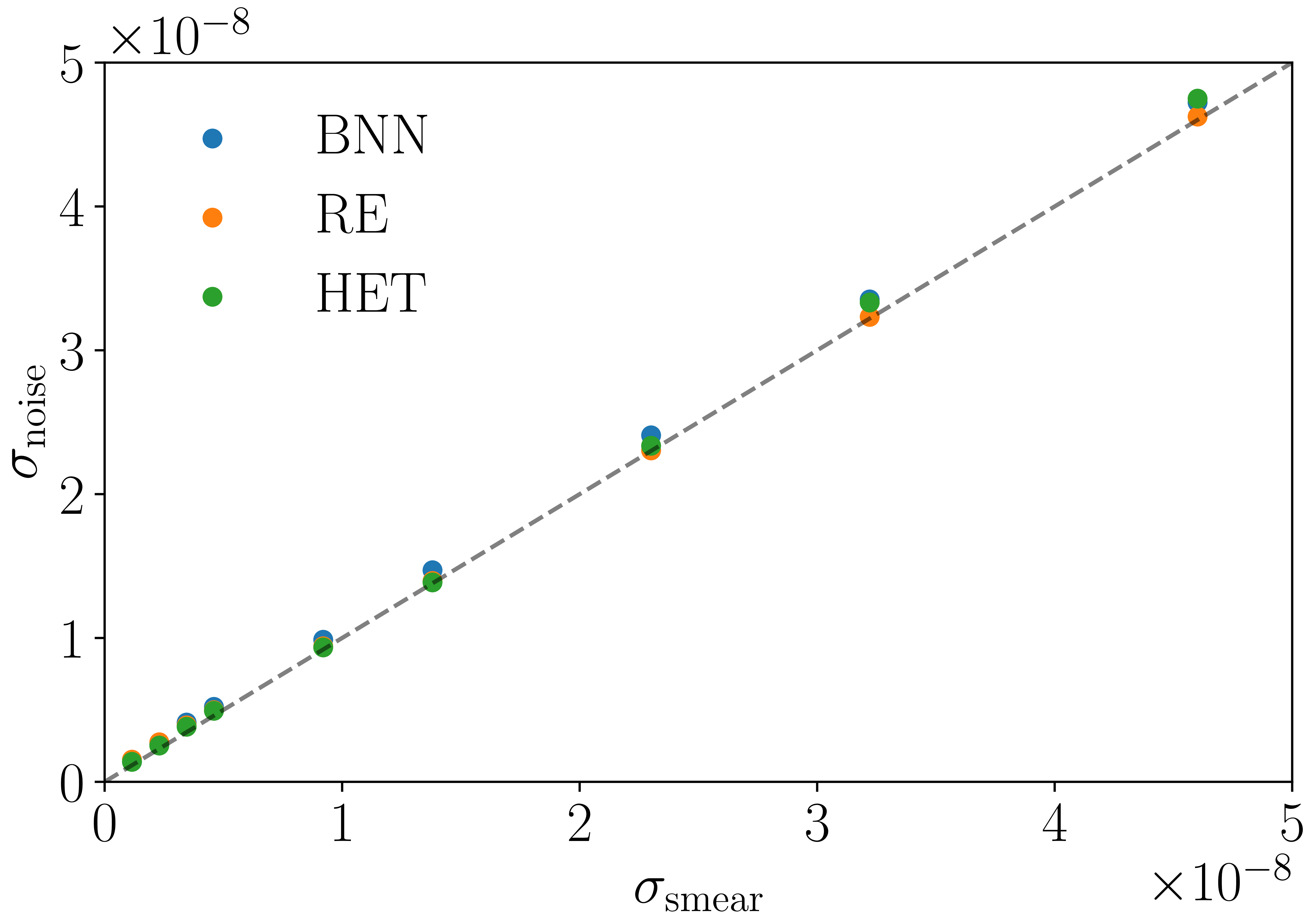
Adding Gaussian noise

$$\sigma_{\text{tot}}^2 = \sigma_{\text{syst},0}^2 + \sigma_{\text{noise}}^2 + \sigma_{\text{stat}}^2$$

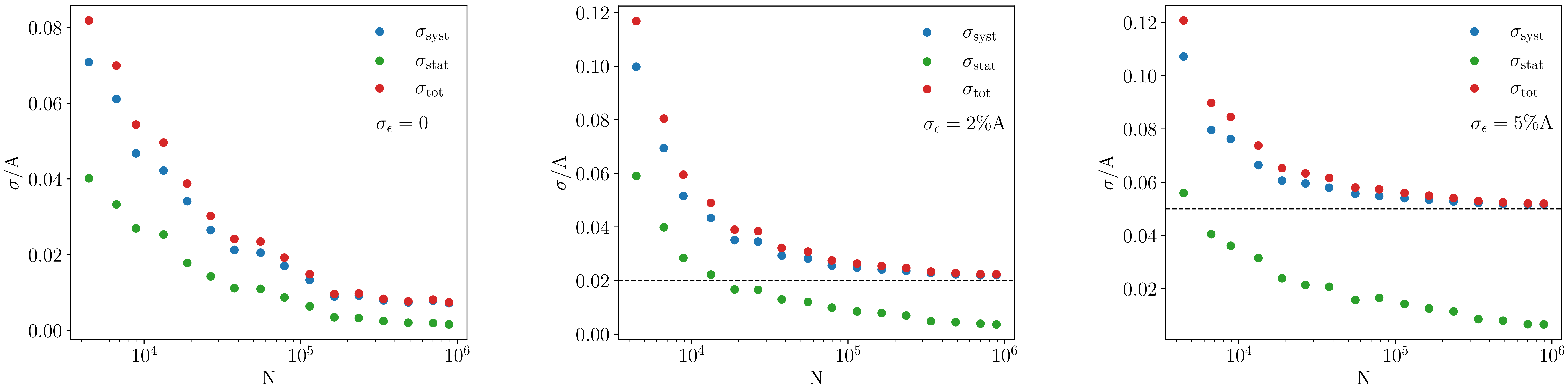
$$\sigma_{\text{smear}} = \{0.25, ..., 10\}\% \times A_{\text{truth}}$$

➡Get additional systematic uncertainty

➡Networks learn noise

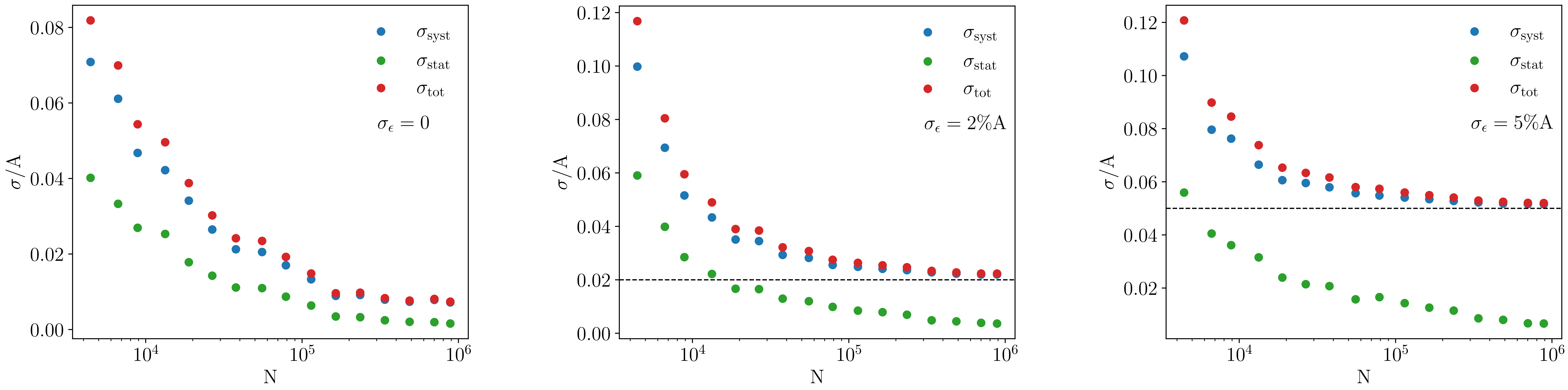


Adding noise to the data



BNN only

Adding noise to the data



BNN only

1. Statistical uncertainty **independent** of noise
2. Systematic uncertainty **plateaus** on noise level

Pull distribution

Problem:

Uncertainties not comparable for different networks

Are uncertainties correctly learned?

Pull distribution

Problem:

Uncertainties not comparable for different networks

Are uncertainties correctly learned?

Use Pull distribution for calibration

Pull distribution

Problem:

Uncertainties not comparable for different networks

Are uncertainties correctly learned?

Use Pull distribution for calibration

$$t_{\text{syst}} = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}}$$

Pull distribution

Problem:

Uncertainties not comparable for different networks

Are uncertainties correctly learned?

Use Pull distribution for calibration

Correctly learned uncertainty should follow a Gaussian shape

$$t_{\text{syst}} = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}}$$

Pull distribution

Pull distribution

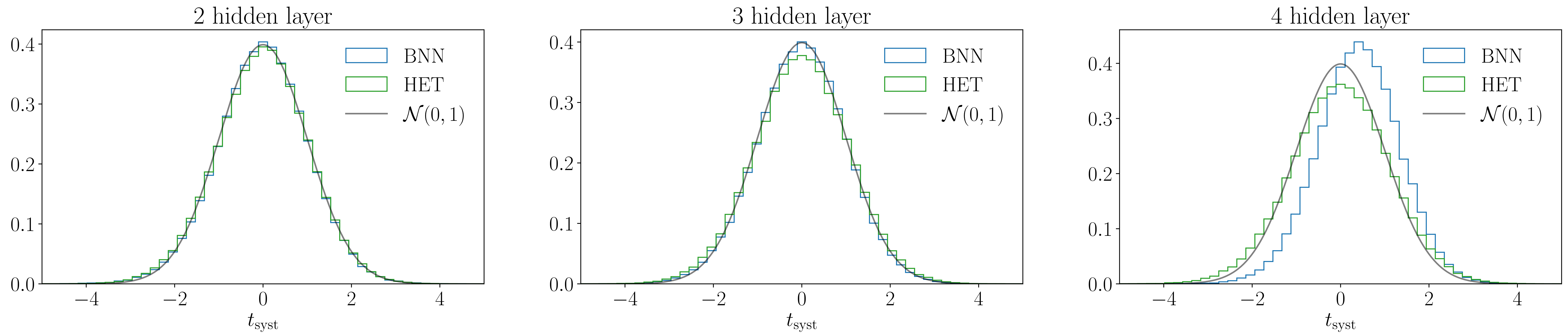
noise	0%	1%	5%	10%
$\langle \sigma_{\text{het}}/A \rangle$	0.005	0.011	0.051	0.104
$\langle \sigma_{\text{syst, BNN}}/A \rangle$	0.006	0.012	0.052	0.103
$\langle \sigma_{\text{syst, RE}}/A \rangle$	0.005	0.011	0.050	0.101

Pull distribution

➡Same relative uncertainty but different pull distribution

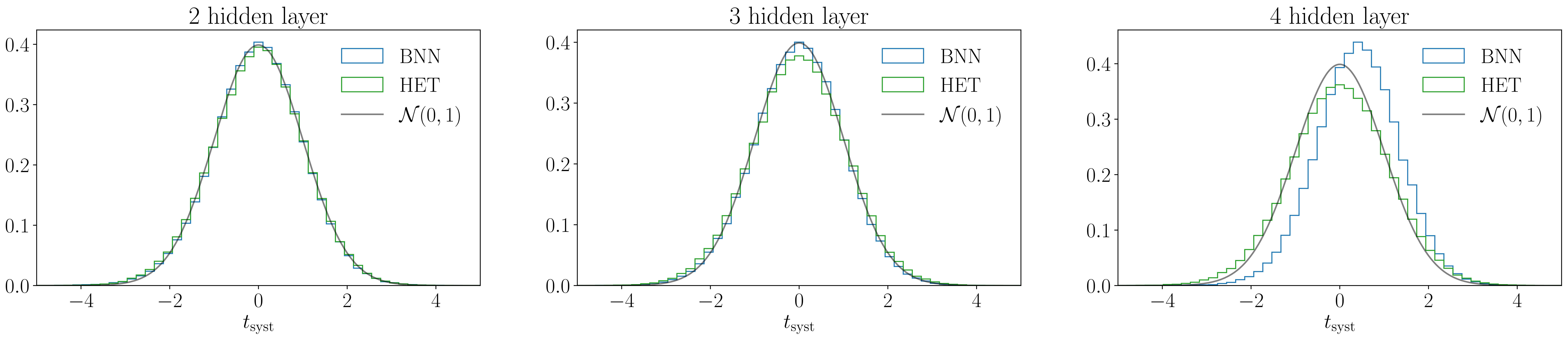
noise	0%	1%	5%	10%
$\langle \sigma_{\text{het}}/A \rangle$	0.005	0.011	0.051	0.104
$\langle \sigma_{\text{syst, BNN}}/A \rangle$	0.006	0.012	0.052	0.103
$\langle \sigma_{\text{syst, RE}}/A \rangle$	0.005	0.011	0.050	0.101

Dependencies on the network size



# hl	1	2	3	4	5	6
$\langle \sigma_{\text{het}}/A \rangle$	0.050	0.010	0.0056	0.0041	0.0038	0.0038
$\langle \sigma_{\text{syst, BNN}}/A \rangle$	0.050	0.012	0.0067	0.0076	0.016	0.017

Dependencies on the network size



➡ Plateau for systematics from network size for het

# hl	1	2	3	4	5	6
$\langle \sigma_{\text{het}}/A \rangle$	0.050	0.010	0.0056	0.0041	0.0038	0.0038
$\langle \sigma_{\text{syst, BNN}}/A \rangle$	0.050	0.012	0.0067	0.0076	0.016	0.017

Reduce systematics

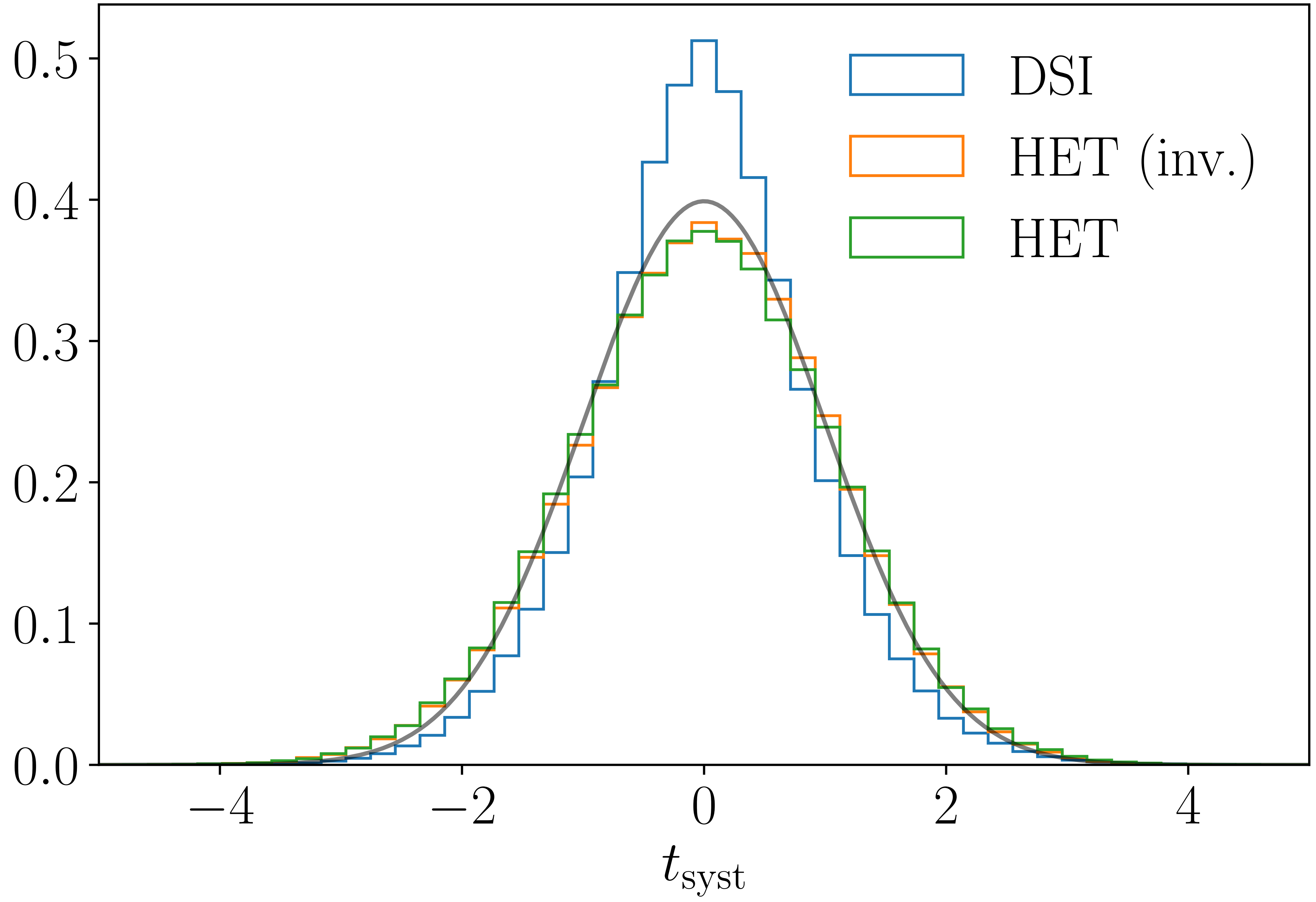
- Gain from different architectures
- Use Deep Set Invariants (DSI)
- Compare to heteroscedastic network with **invariant preprocessing**

network	$\langle \sigma / A \rangle$
HET	0.0056
HET (inv.)	0.00098
DSI	0.000068

Reduce systematics

- Gain from different architectures
- Use Deep Set Invariants (DSI)
- Compare to heteroscedastic network with **invariant preprocessing**

network	$\langle \sigma / A \rangle$
HET	0.0056
HET (inv.)	0.00098
DSI	0.000068



Conclusion and Outlook

1. Networks are able to learn Gaussian noise

2. Relative systematic uncertainties are similar but the pull distribution might look different

3. Reduce systematics by either changing the network size or include symmetries

Next: Look at statistical pull distribution, deeper look into the DSI

Thank you for your attention!