Quantifying Uncertainties in Weight-Parameterized Residual Neural Networks

Khachik Sargsyan (SNL),
Oscar Diaz-Ibarra (SNL), Javier Murgoitio-Esandi (USC),
Joshua Hudson (U Arkansas), Marta D'Elia (Stanford and Pasteur Labs), Habib Najm (SNL)

Sandia National Laboratories

SIAM UQ, Trieste, Italy
March 1, 2024

Outline

- UQ for NNs: review and state of the art
  - Needed for SciML workflows: active learning, comp. design...
  - Loss landscape perspective, challenges, metrics
- Weight parametrization in Residual NNs (ResNets)
  - Reduces generalization gap
  - Enables easier UQ
- QUiNN: ongoing work and software plug

Probabilistic NN == Bayesian NN

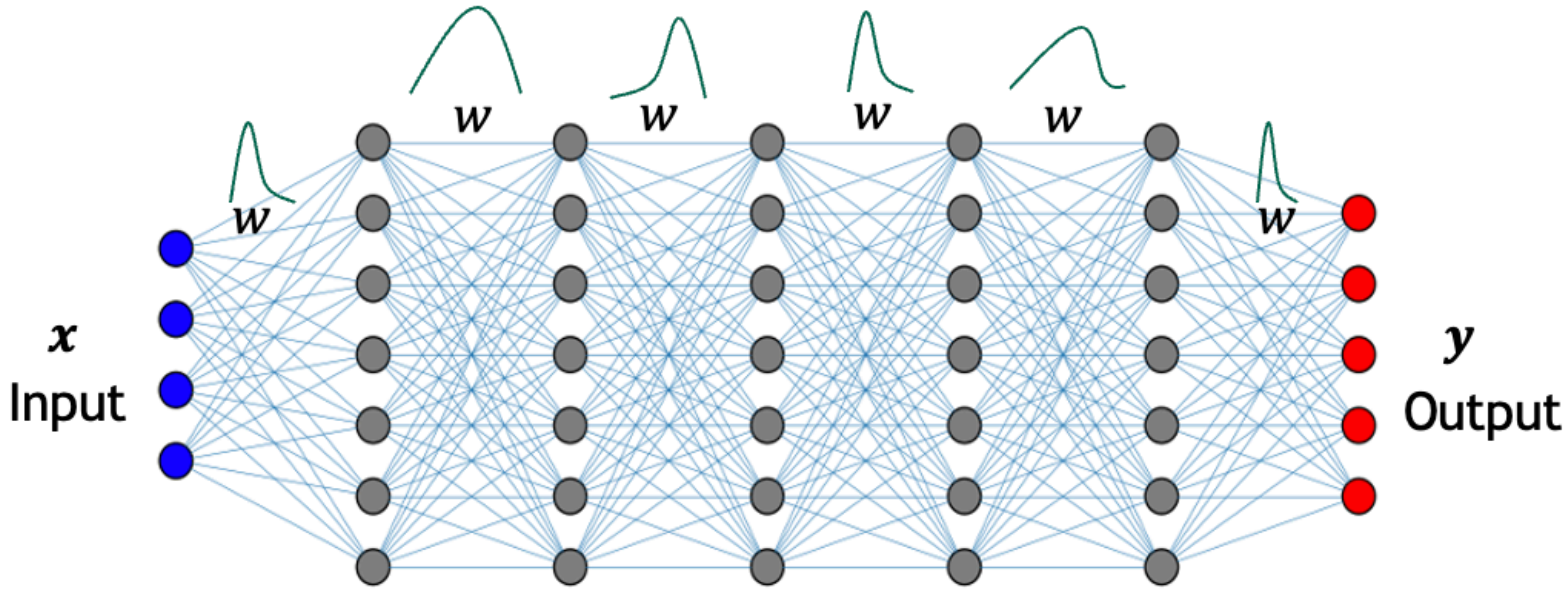
Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence". Nature, 2015

"Nearly all approaches to probabilistic programming are ***Bayesian*** *since it is hard to create other coherent frameworks for automated reasoning about uncertainty"*

- Bayesian NN methods have been around since 90s *[MacKay, 1992; Neal, 1996]*
- Full Bayesian treatment was infeasible back then….
  - … and still is, generally, not industry-standard by any means.

UQ-for-NN: Bayesian perspective

Training for NN weights reformulated as a Bayesian inference problem



Posterior Prior

$$p(w \mid y) \propto p(y \mid w)\, p(w)$$

Likelihood

$$\propto \exp\left(-\frac{||y - f_w(x)||^2}{2\sigma^2}\right) \exp\left(-\frac{||w||^2}{2\lambda^2}\right)$$

Negative Log-Posterior $\simeq a||y - f_w(x)||^2 + b||w||^2 \simeq$ Training Loss Function

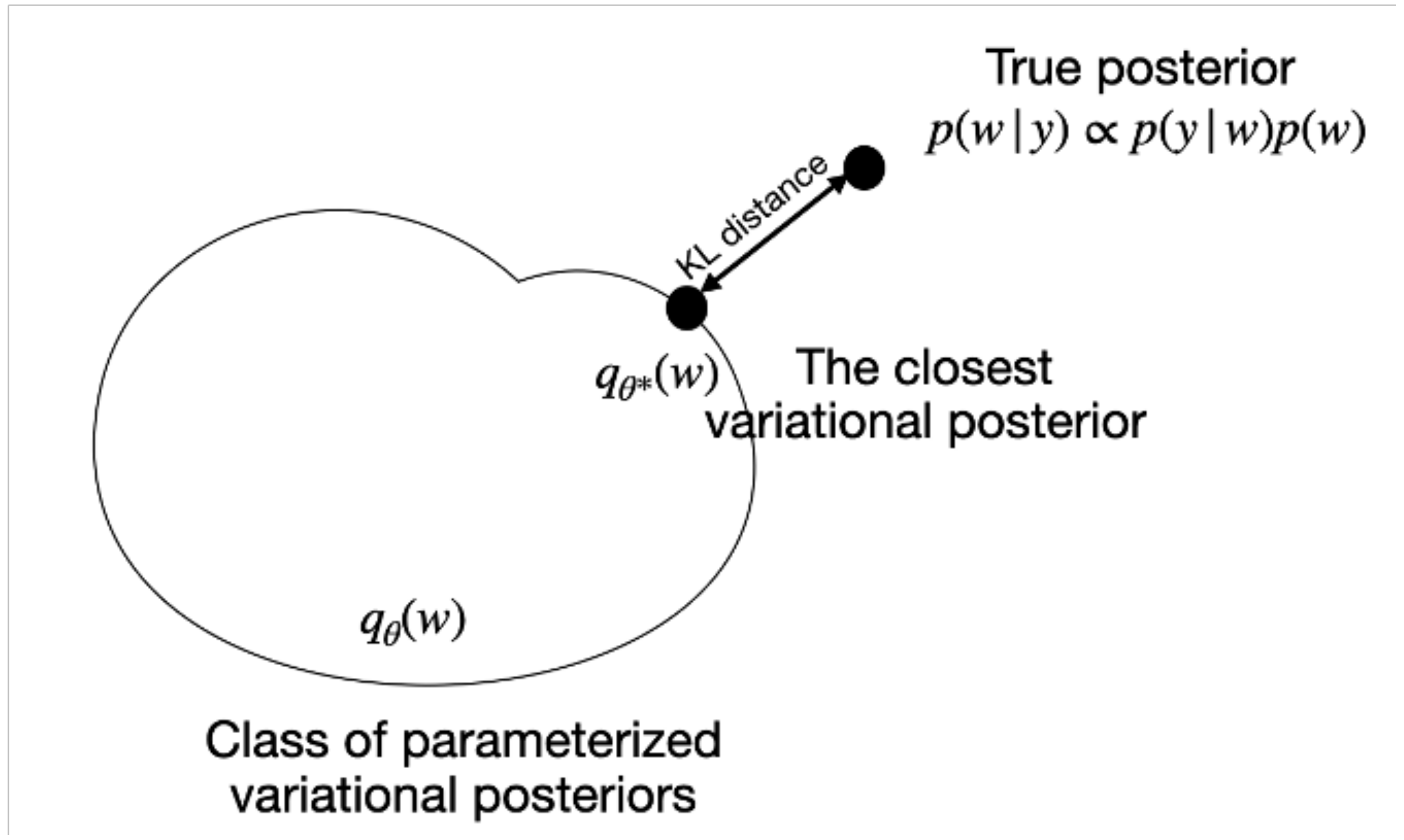
- ✓ Markov chain Monte Carlo (MCMC) sampling; Hamiltonian MC *[Levy, 2018]*
- ◉ Tuning is an art: essentially infeasible outside academic examples

UQ-for-NN: variational methods

- Bayes by Backprop *[Blundell, 2015]*
  - has become mainstream in ML literature
  - also called BNN

  - Mean-field VI (i.e. i.i.d. normal variational class)
  - Reparameterization trick
  - Gaussian mixture prior: wide and narrow
  - Variational st.dev. $\sigma = ln(1 + e^{\rho})$

- SVI, ADVI, BBVI, BBBVI, CCVI, CATVI, ….



◉ Typically underestimates predictive uncertainty
◉ Restricted to variational class
◉ Hard to train

UQ-for-NN: approximate methods

- **Probabilistic backprop, or PBP** *[Hernandez-Lobato, 2015]*
  - Layer-to-layer updates from $\mathcal{N}(\mu, \sigma^2)$ to $\mathcal{N}(\mu_{new}, \sigma^2_{new})$
  - Deriving back propagation formulas for this update
  - $\mu, \sigma^2 \rightarrow \mu_{new}, \sigma^2_{new}$ updates similar to PC propagation (1st order Gauss-Hermite PC)

◉ Did not really lift off

◉ Original implementation in Theano

- **Laplace methods**: *[Ritter, 2018, Daxberger, 2021]*

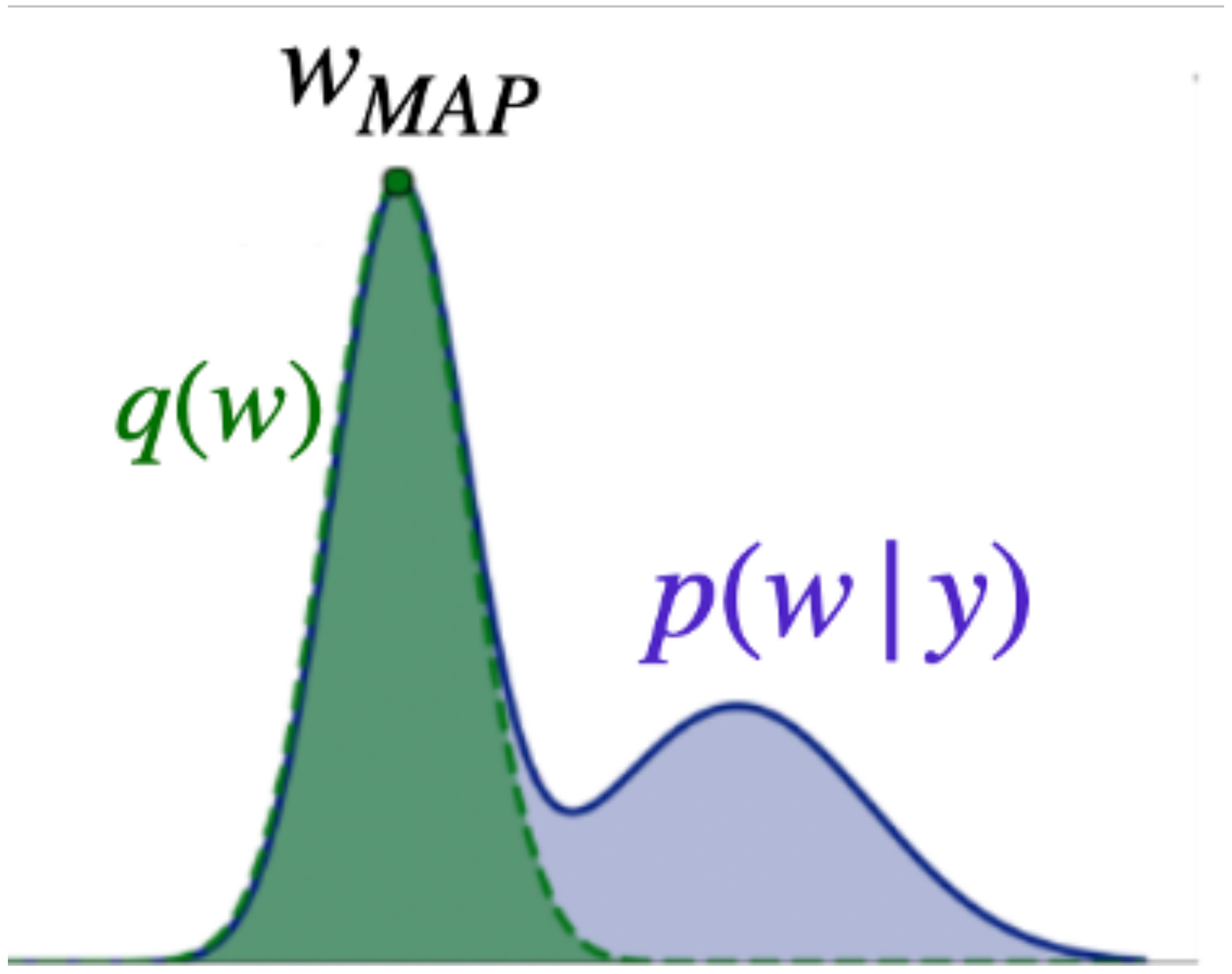
  ✓ Relies on Gaussian apprx near maximum;

  ✓ Can be generalized to GMM

  ◉ Good only locally

  ◉ Hessian computation challenging

  ◉ Fails to explore the full posterior



UQ-for-NN: other (more empirical) methods

- **Ensembling methods:** work surprisingly well!
  - ✓ Deep Ensembles *[Lakshminarayanan, 2017];*
  - ✓ Interpreting ensembles from Bayesian perspective *[Garipov, 2018; Fort, 2019]*
  - ✓ Randomized MAP Sampling (anchored ensembles) *[Pearce, 2020]*
  - ✓ MC-Dropout *[Gal, 2015]*
  - ✓ Stochastic Weight Averaging – Gaussian (SWAG) *[Maddox, 2019]*:shipped w PyTorch1.6
  - ✓ Delta-UQ *[Anirudh, 2021]*,
  - ✓ AutoDEUQ *[Egele, 2022].*
  - ◉ Often little theoretical backing
  - ◉ Too expensive, albeit parallelizable

- **Direct learning of predictive RV**
  - ✓ Distance-based methods *[Postels, 2022],*
  - ✓ DEUP *[Lahlou, 2023]*
  - ✓ AVUC *[Krishnan, 2020]*.

- **Other**
  - ✓ Information-bottleneck UQ *[Guo, 2023]*,
  - ✓ Conformal UQ *[Hu, 2022]*,
  - ✓ Bayesian Last Layer *[Watson, 2021],*
  - ✓ TAGI *[Goulet, 2021].*

Challenges of UQ-for-NN

✓ Complicated posterior distribution (loss landscape):

- invariances and symmetries: permuting some weights leads to the same loss,
- multimodality: multiple local minima in the weight space,
- “ridges”: low-d manifolds with same or similar loss.

✓ Prior on weights hard to elicit/interpret/defend

- what does a uniform/gaussian prior on weight matrix elements mean?
- perhaps a prior is needed in the ‘matrix’-space, or…
- driven by outputs, or physics-constraints.

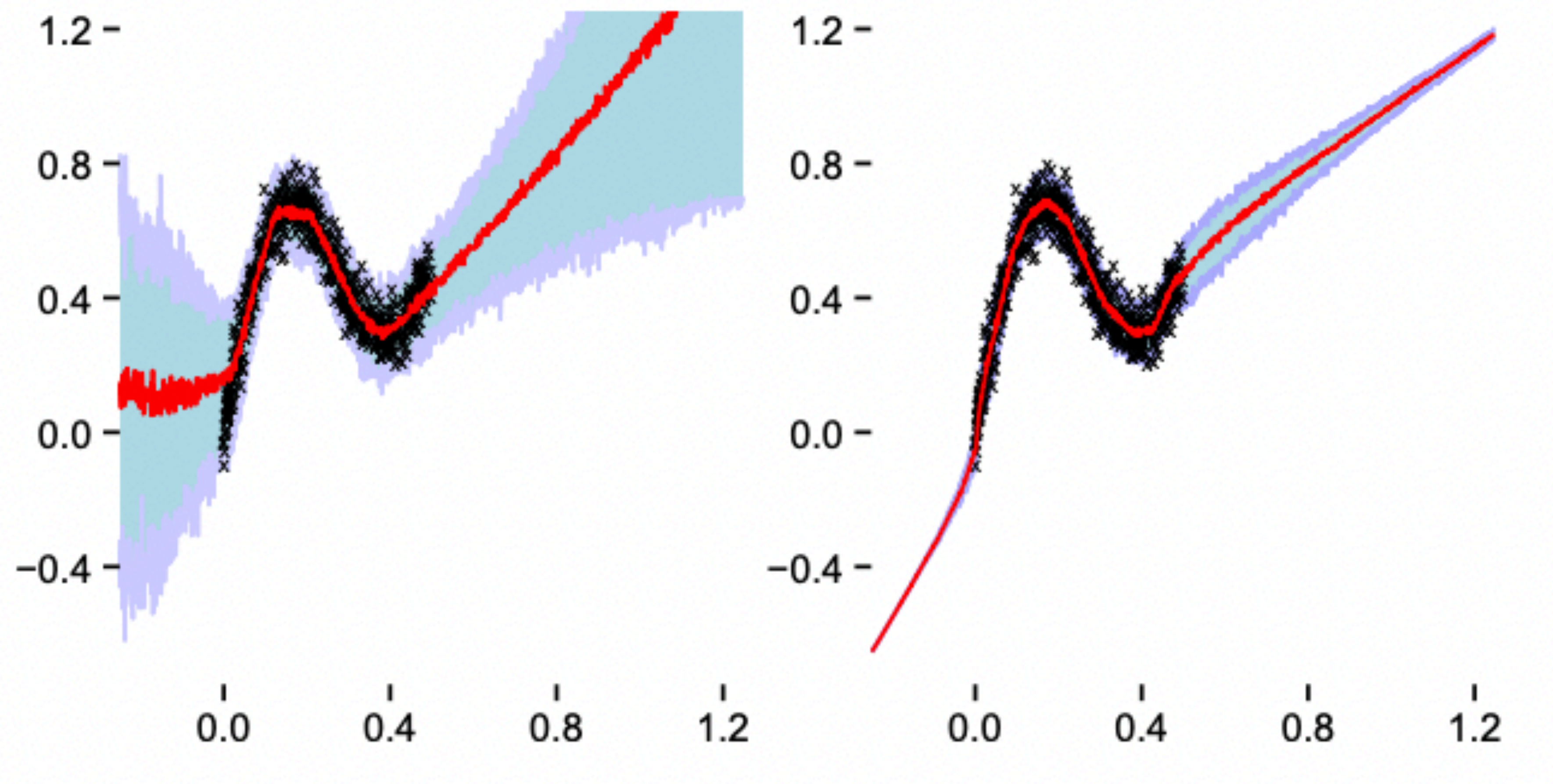
✓ Large number of weights:

- scales linearly with depth and quadratically with width,
- hard to visualize the high-d surface.

How to measure if uncertainty estimate is correct?

✓ Still a lot of eyeballing and 1d fit examples,

✓ Striving to match a GP

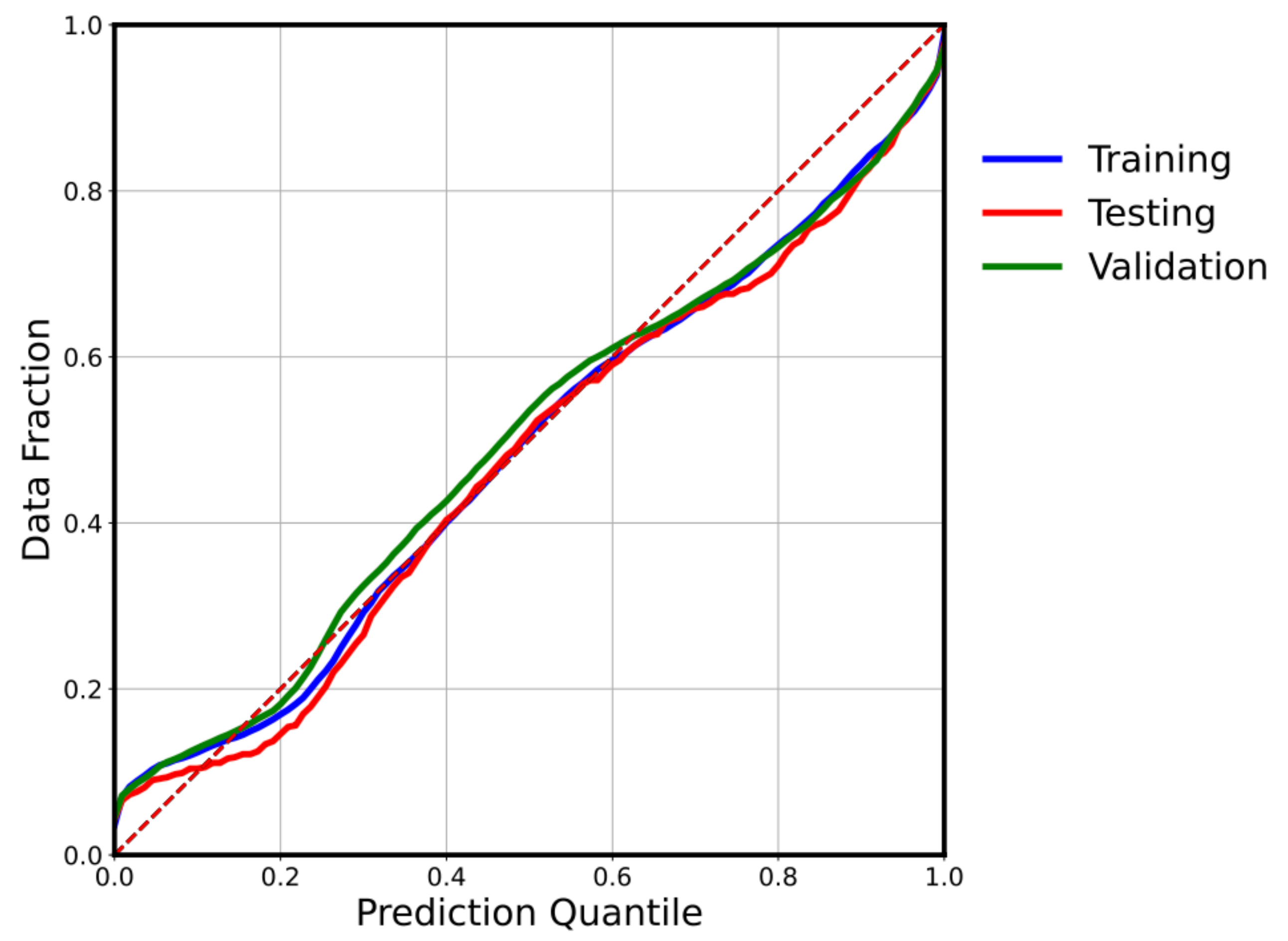


✓ Benchmarking efforts are picking up:

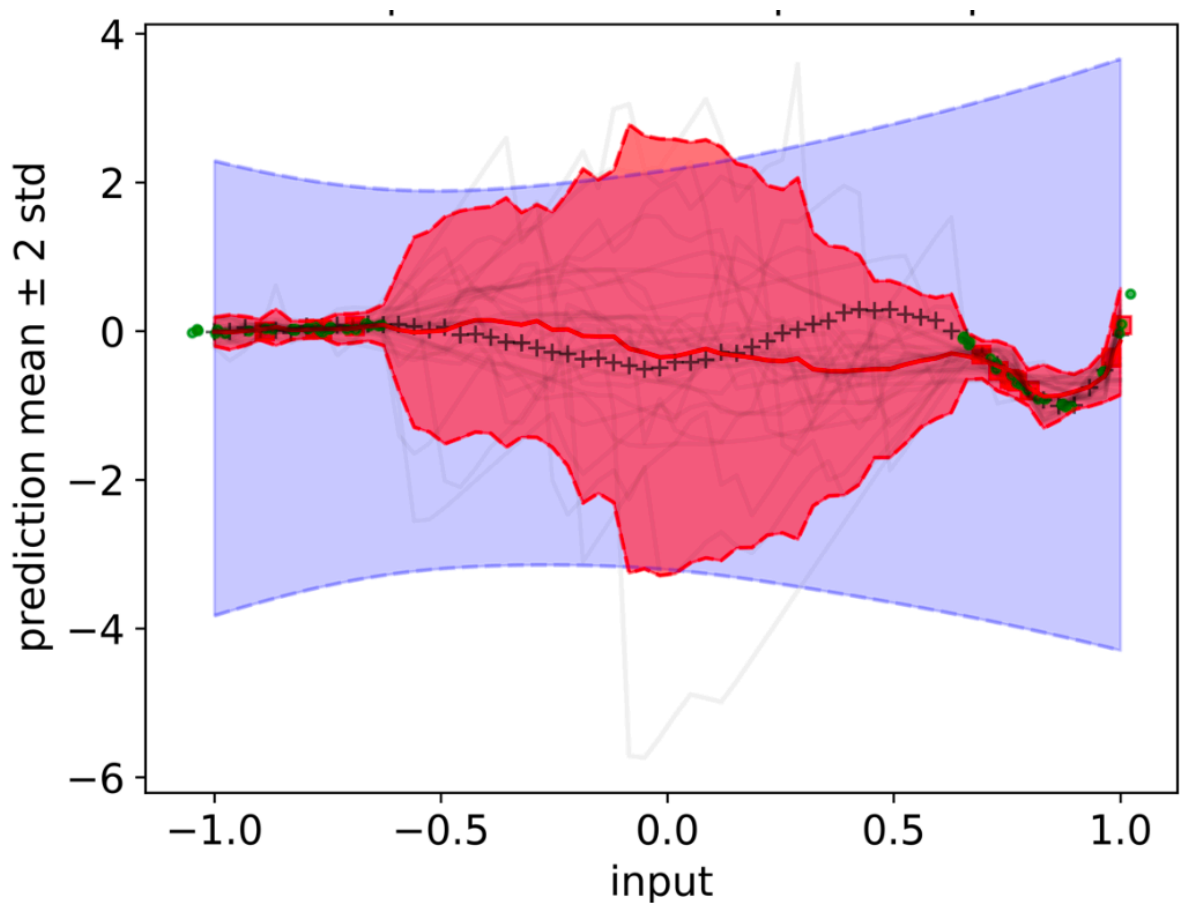
- UCI Dataset, both regression and classification
- Recent work specific to Bayesian NN

[Yao, 2019; Navratil, 2021; Nado, 2021; Staber, 2022; Basora, 2023]

Uncertainty-Accuracy Plot



Posterior predictive with no data —> Prior predictive



Loss Landscape Perspective

- Visualization of loss surface is key to help understand and characterize NN performance *[Li, 2018; Garipov, 2018; Fort, 2019; Yang, 2021]*,

- Incorporating prior knowledge should regularize the loss/log-posterior landscapes, making them more amenable to sampling and analysis.
- This means both:
  - *soft* regularization (like PINN) and
  - *hard* architectural changes
    - physics-driven rewiring (invariance, symmetries, positivity, feature extraction),
    - numerical convenience (**ResNet**/NODE, **weight reparameterization**, layer/batch normalization).

ResNet/NODE in regression setting

ResNet (discrete)

$$\begin{cases} \boldsymbol{x}_1 = \boldsymbol{x} + \boldsymbol{\alpha}_0\boldsymbol{\sigma}(\boldsymbol{W}_0\boldsymbol{x}_0 + \boldsymbol{b}_0) \\ \vdots \\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\alpha}_n\boldsymbol{\sigma}(\boldsymbol{W}_n\boldsymbol{x}_n + \boldsymbol{b}_n) \\ \vdots \\ \boldsymbol{y} = \boldsymbol{x}_{L-1} + \boldsymbol{\alpha}_{L-1}\boldsymbol{\sigma}(\boldsymbol{W}_{L-1}\boldsymbol{x}_{L-1} + \boldsymbol{b}_{L-1}) \end{cases}$$

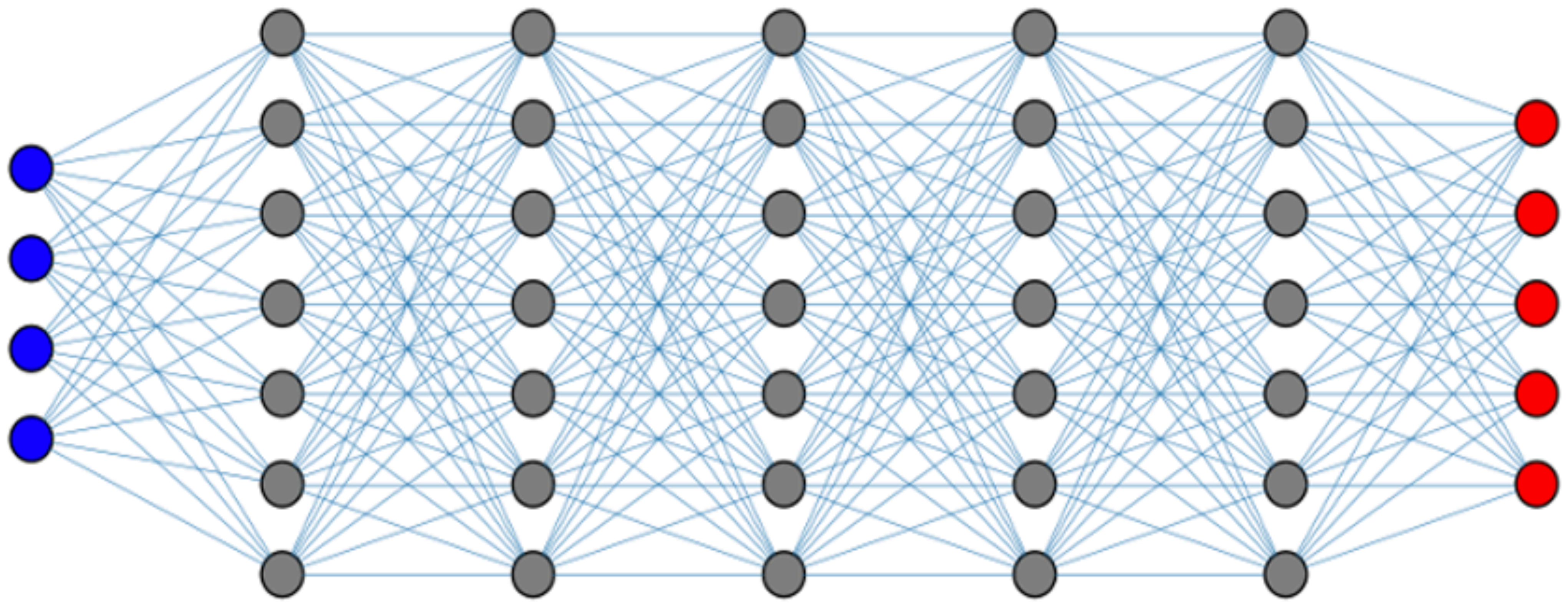
Neural ODE (continuous)

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{\sigma}(\boldsymbol{W}(t)\boldsymbol{x} + \boldsymbol{b}(t))$$

$$\boldsymbol{x}(0) = \boldsymbol{x} \qquad \boldsymbol{x}(T) = \boldsymbol{y}$$

$\boldsymbol{x} = \boldsymbol{x}_0$

Input



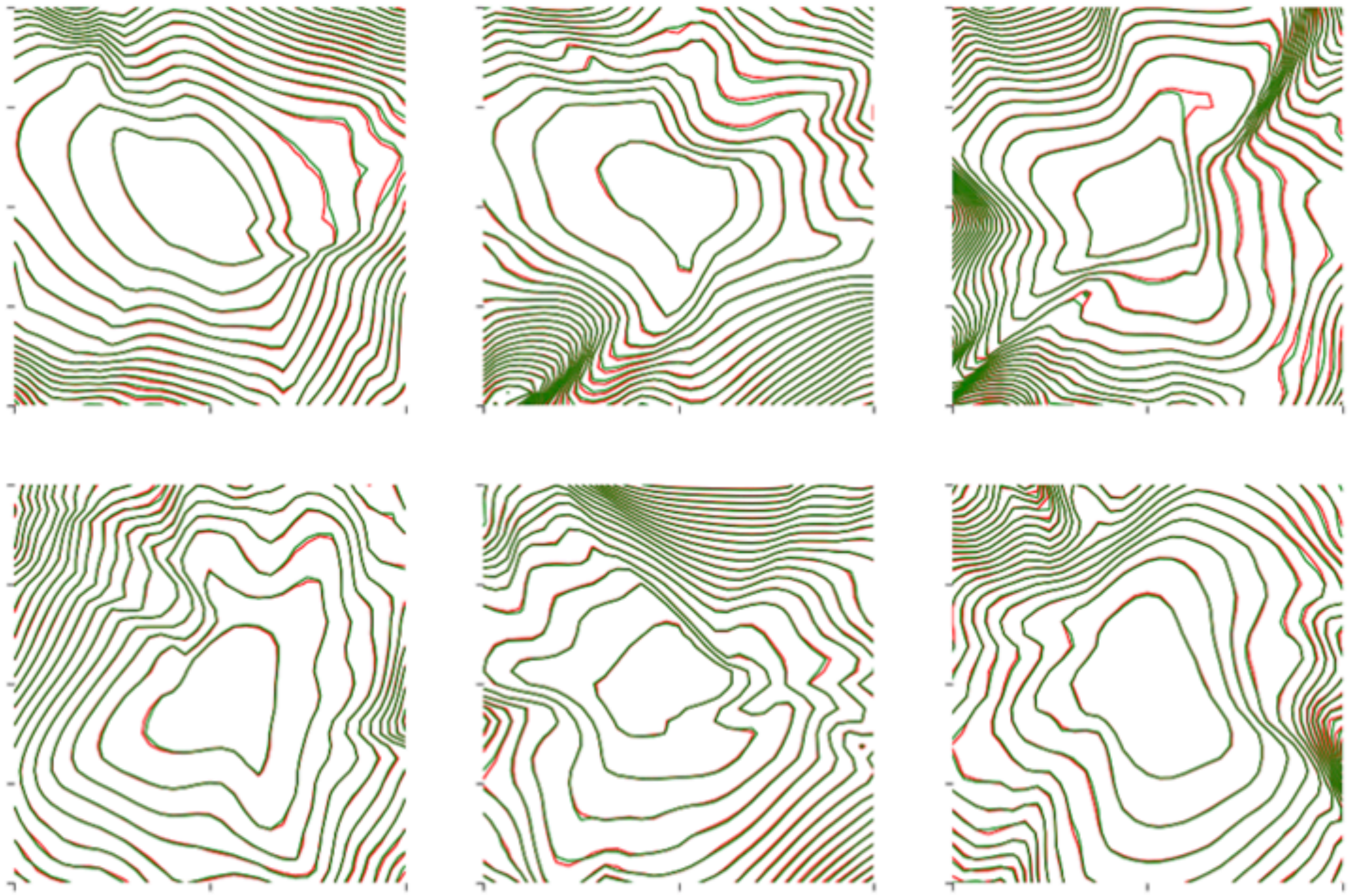
$\boldsymbol{y} = \boldsymbol{x}_T$

Output

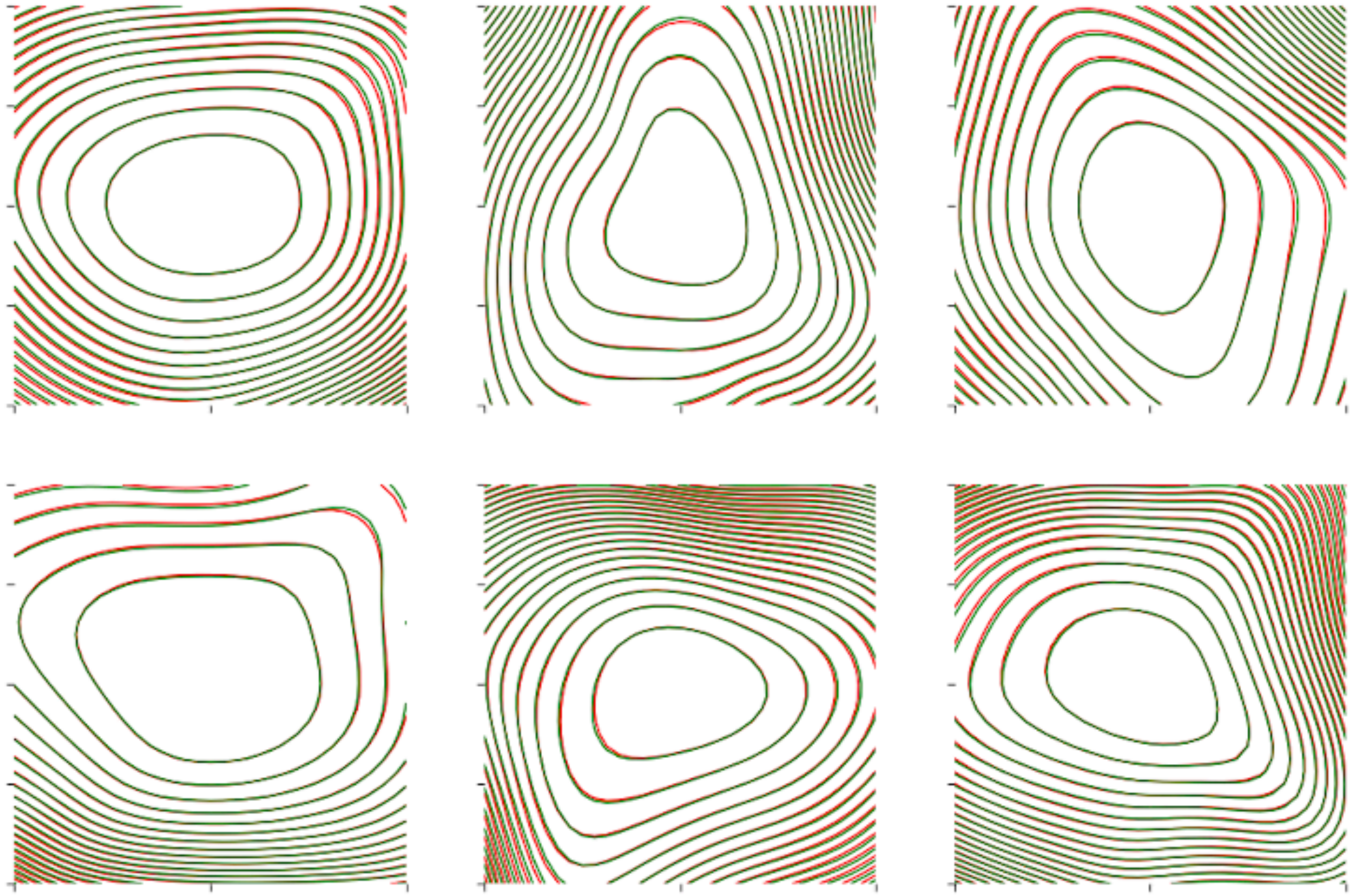
[E, 2017; Chen, 2018; Ruthotto, 2018]

ResNet shortcuts regularize loss landscape

Conventional MLP: $x_{n+1} = \sigma(W_n x_n + b_n)$



ResNet: $x_{n+1} = x_n + \sigma(W_n x_n + b_n)$

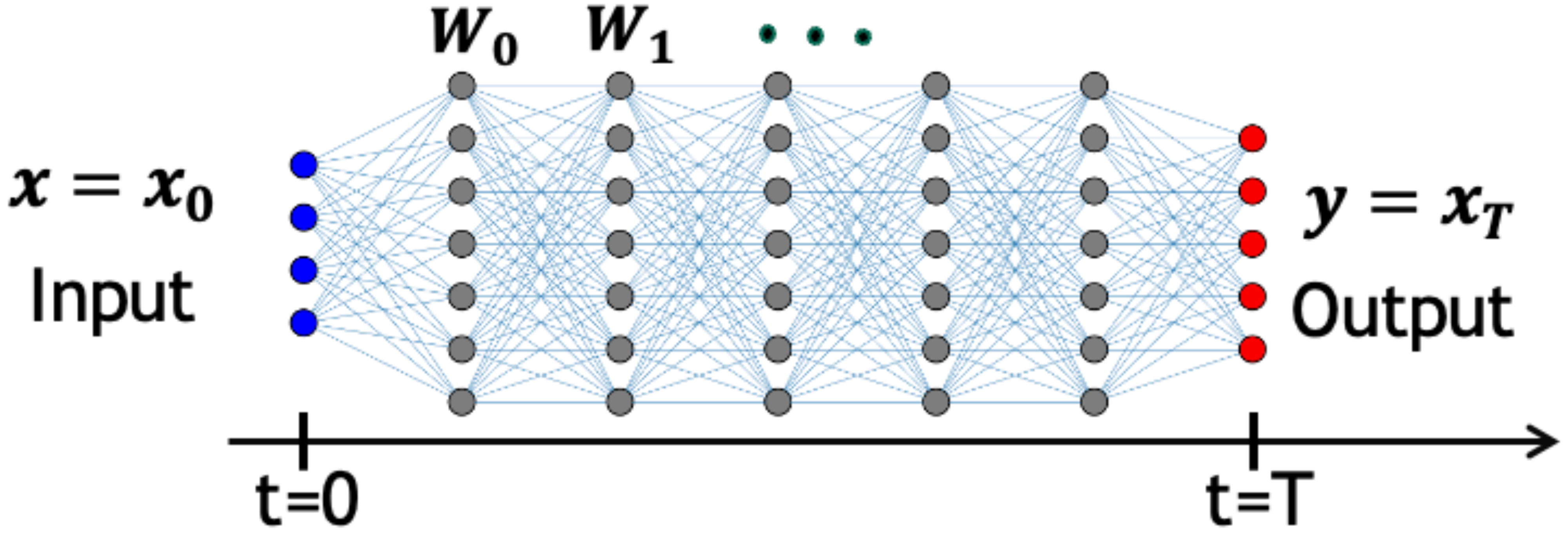


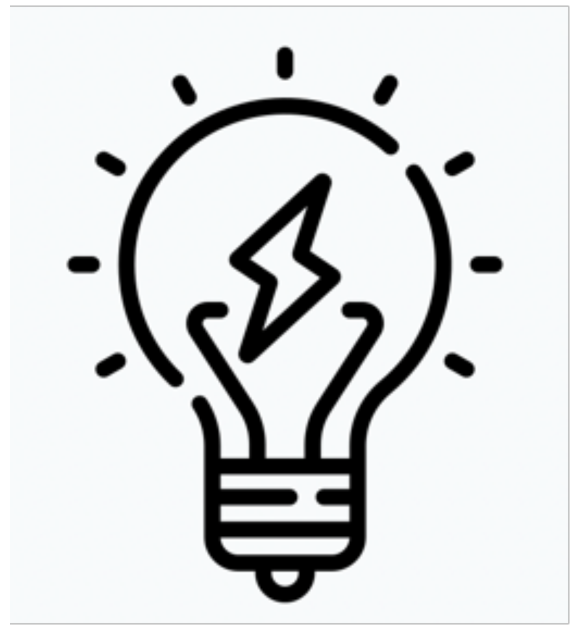
See *[Lee, 2017]* for a more comprehensive study.

Weight Parameterization inspired by NODE analogy

Neural ODE: $$\frac{dx}{dt} = \sigma(W(t)x + b(t))$$

ResNet: $$x_{n+1} = x_n + \sigma(W_n x_n + b_n)$$





Parameterize weight matrices with respect to time (aka depth)

$W(t;\theta)$ and train for θ's.

Weight Parameterization as a regularization tool

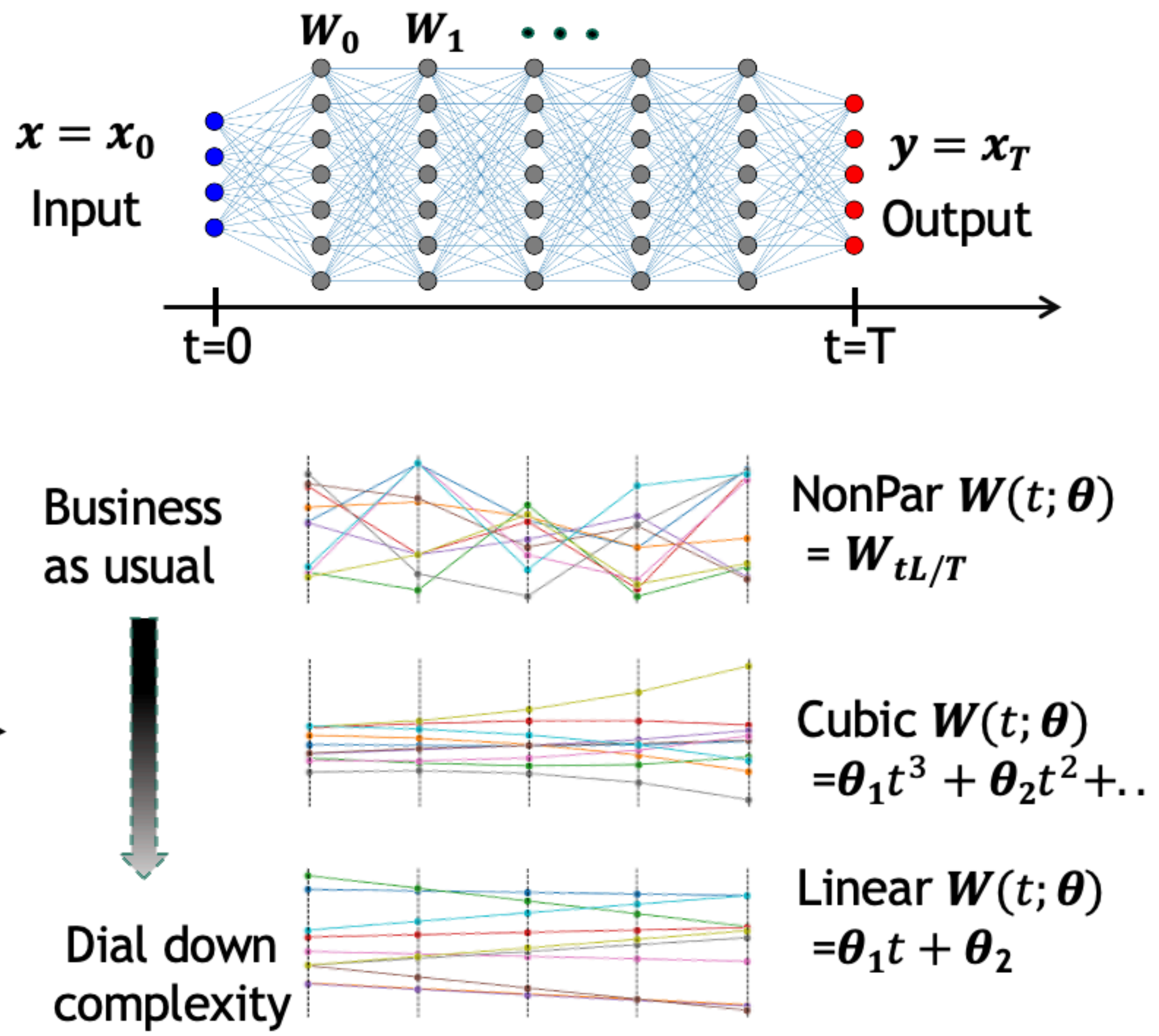
ResNet: $x_{n+1} = x_n + \alpha_n \sigma(W_n x_n + b_n)$

Training for weight matrices $W_0, W_1, \ldots$

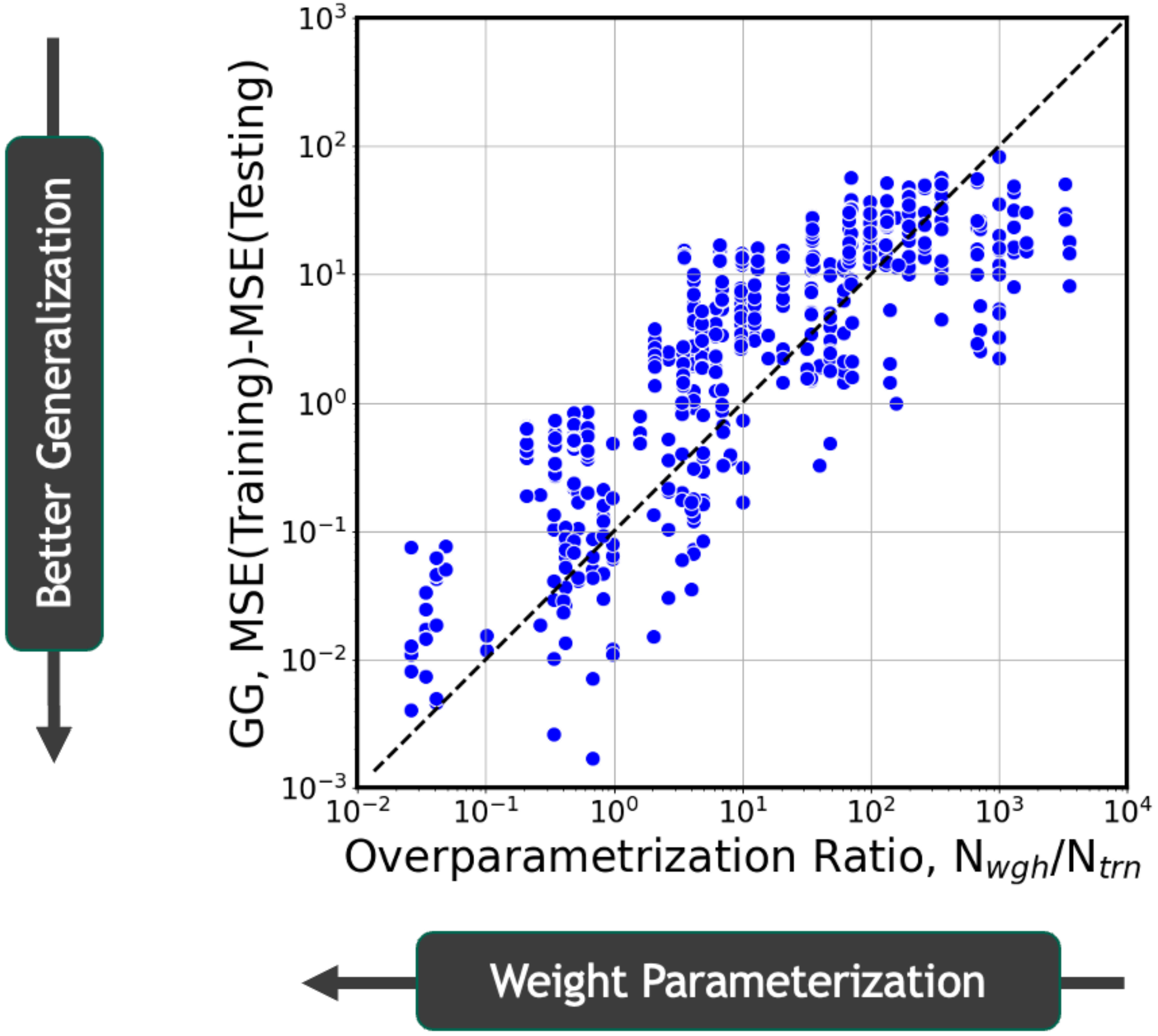
Heavily overparameterized, does not generalize well

Parameterize $W(t;\theta)$ and train for θ's.

Parameterization of weight functions reduces capacity and improves generalization



Weight Parameterization improves generalization

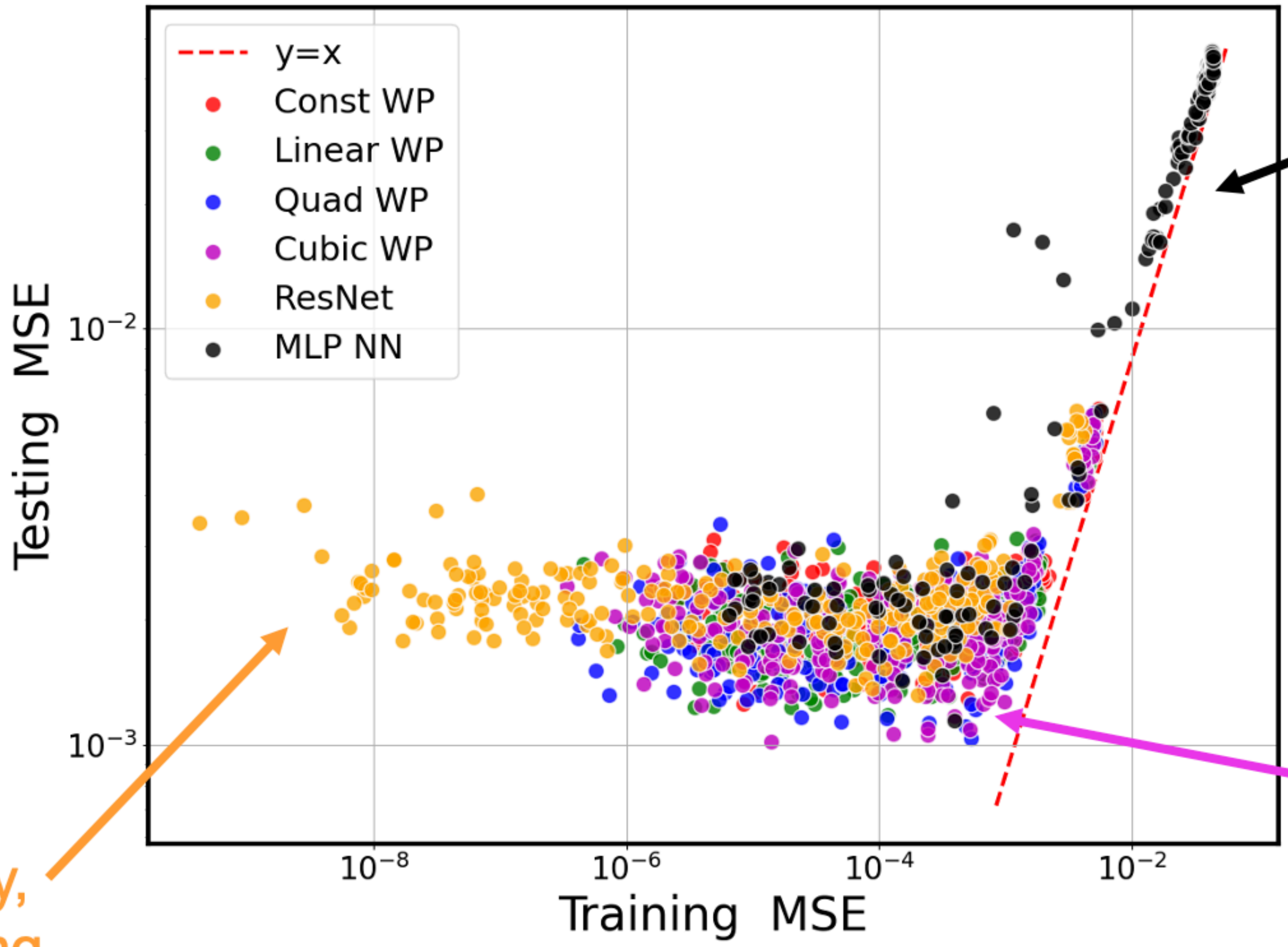


- Generalization Gap correlates with overparameterization

- Weight-parameterized ResNets reduce Generalization Gap

Each dot is a training run with varying weight parameterization functions

Weight Parameterization improves accuracy



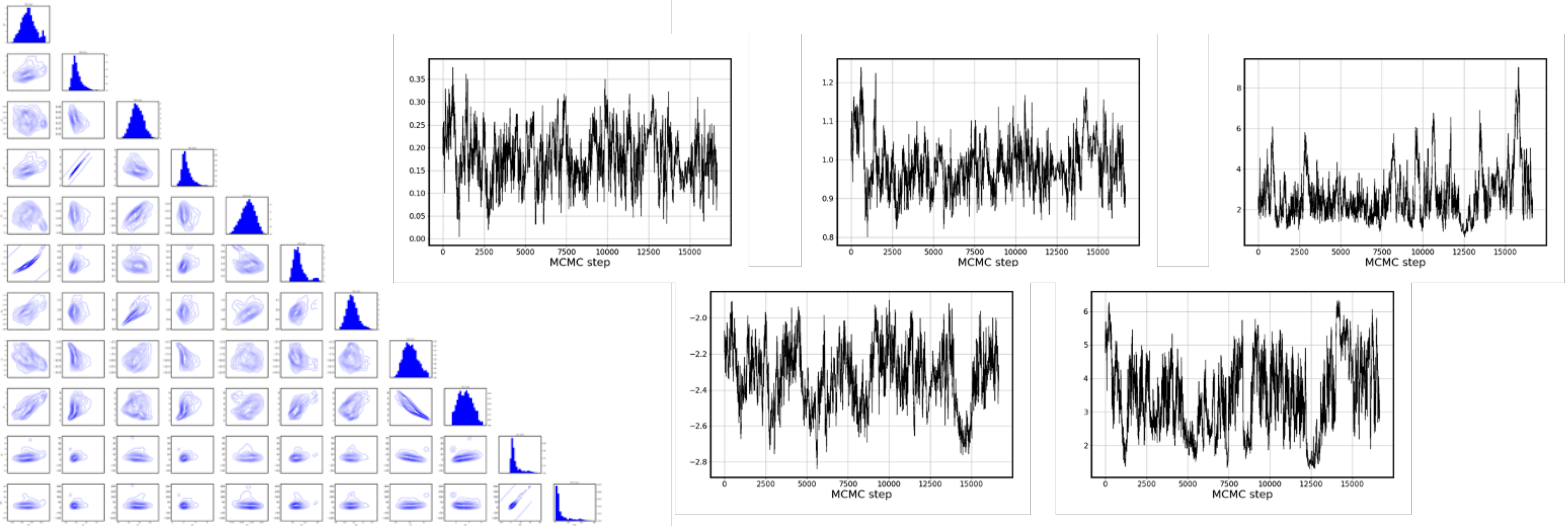
MLP:
bad accuracy

WP:
perfect tradeoff

ResNet:
ok accuracy,
but overfitting

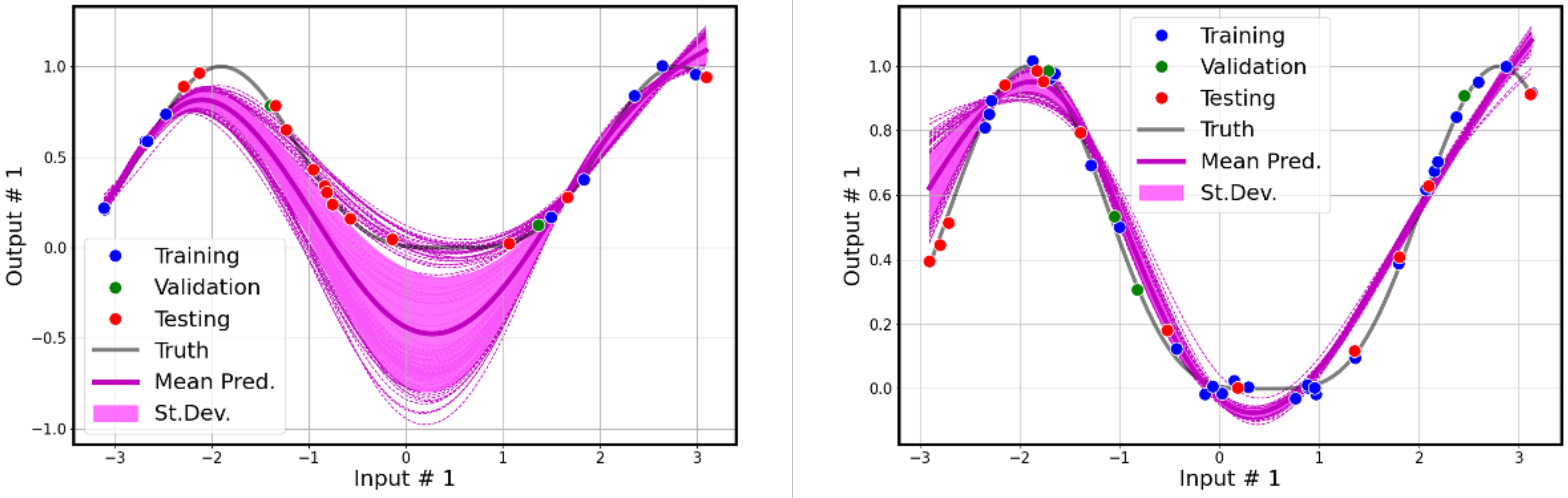
WP ResNet enables UQ

- Number of parameters in ResNets, as well as MLPs, **grows with linearly depth**.
- Number of parameters in weight-parameterized ResNets is **independent of depth**.
- We can easily achieve regimes with manageable MCMC dimensionality and posterior PDFs that out-of-box MCMC methods can easily sample.



WP ResNet enables UQ

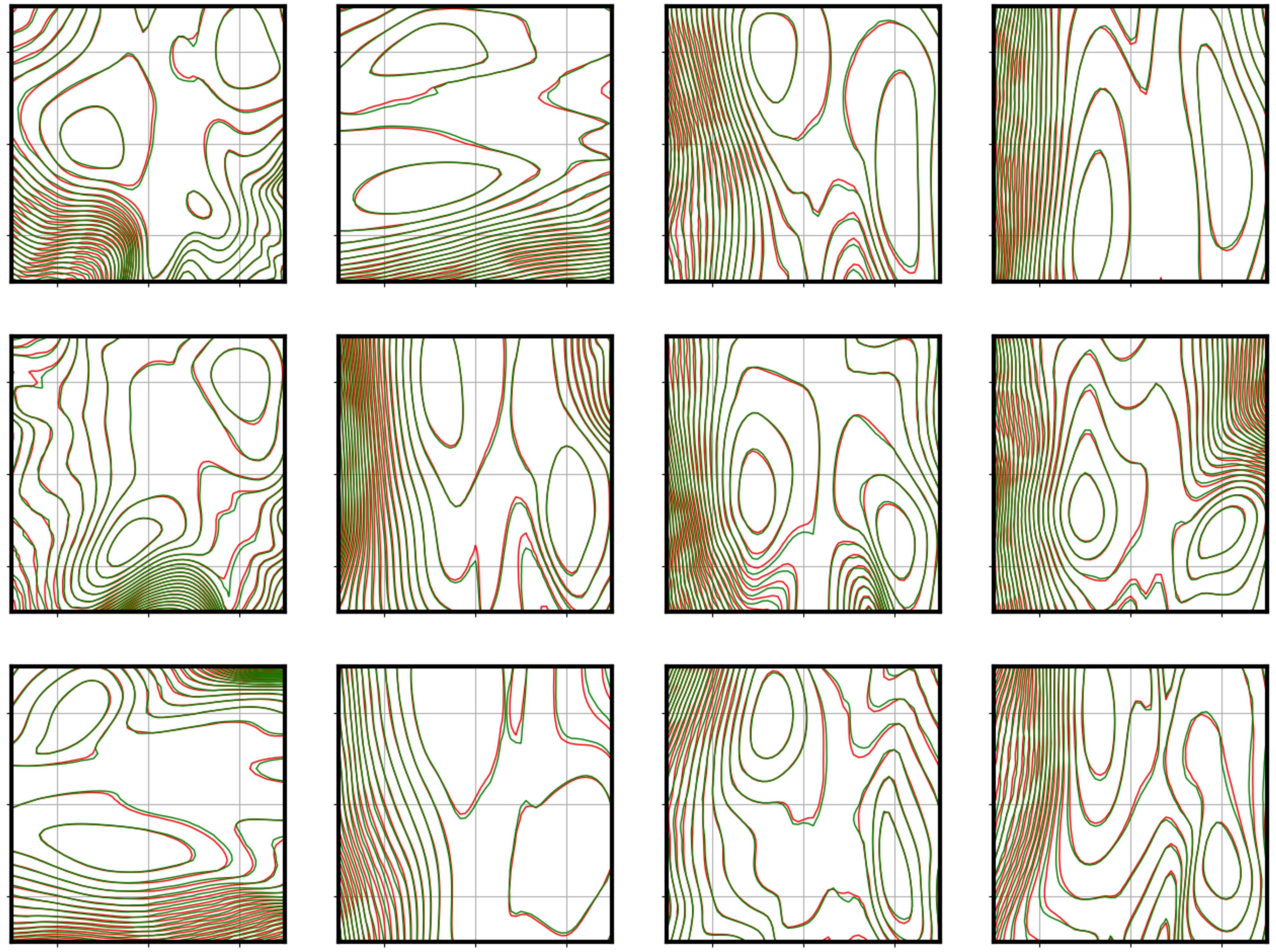
- Number of parameters in ResNets, as well as MLPs, **grows linearly with depth**.
- Number of parameters in weight-parameterized ResNets is **independent of depth**.
- We can easily achieve regimes with manageable MCMC dimensionality and posterior PDFs that out-of-box MCMC methods can sample.



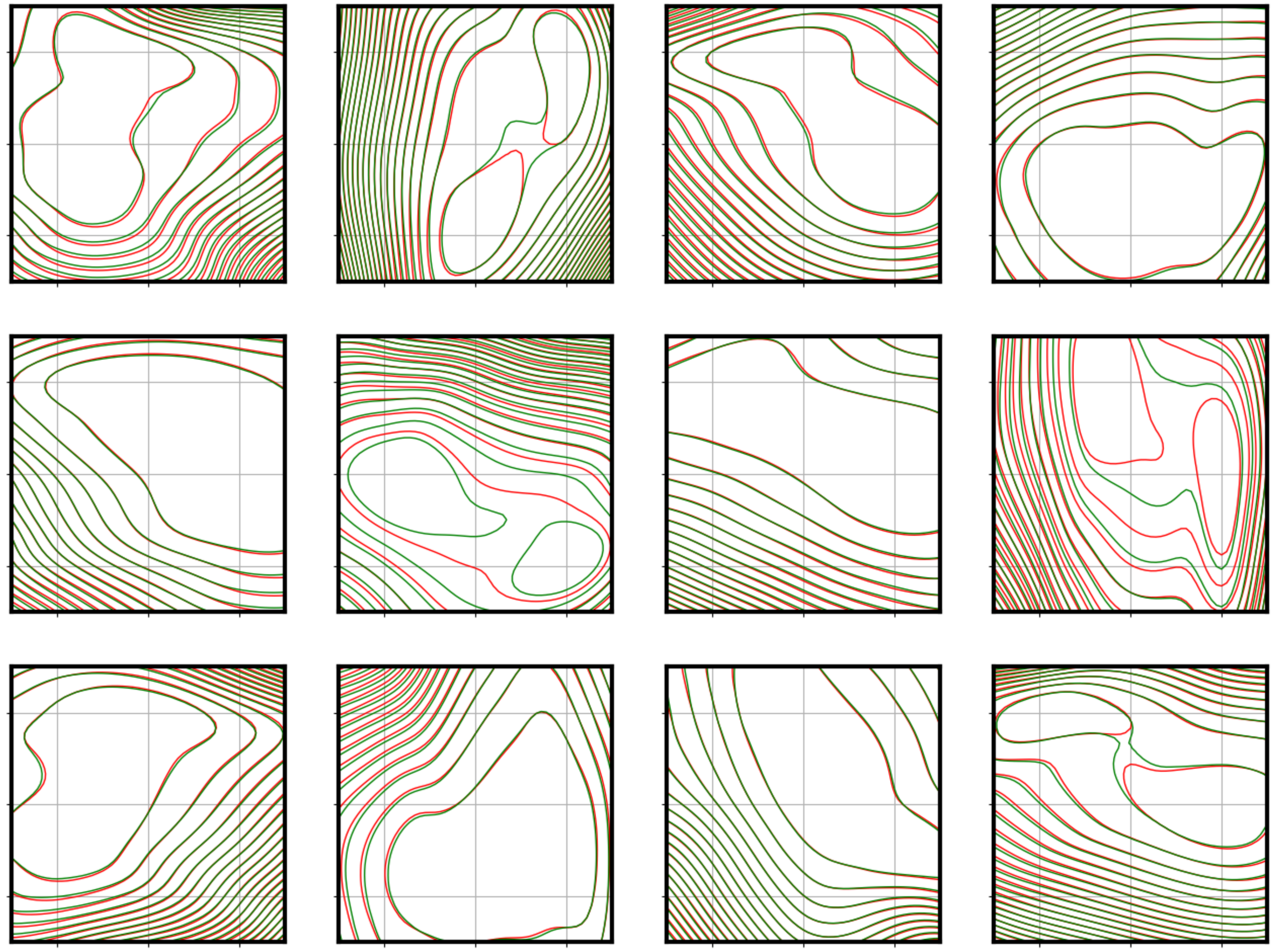
WP ResNet enables UQ

- Number of parameters in ResNets, as well as MLPs, **grows linearly with depth**.
- Number of parameters in weight-parameterized ResNets is **independent of depth**.
- We can easily achieve regimes with manageable MCMC dimensionality and posterior PDFs that out-of-box MCMC methods can sample.

MLP



WP-ResNet



QUiNN: github.com/sandialabs/quinn

Deterministic → **Probabilistic**

torch.nn.module → **wrapper(torch.nn.module)**

Usage: →

```
uqnet = MCMC_NN(nnet)
```

```
class MCMC_NN(QUiNNBase):
    def __init__(self, nnmodule, verbose=True):
        super(MCMC_NN, self).__init__(nnmodule)
        self.verbose = verbose
```

MCMC

```
uqnet = VI_NN(nnet)
```

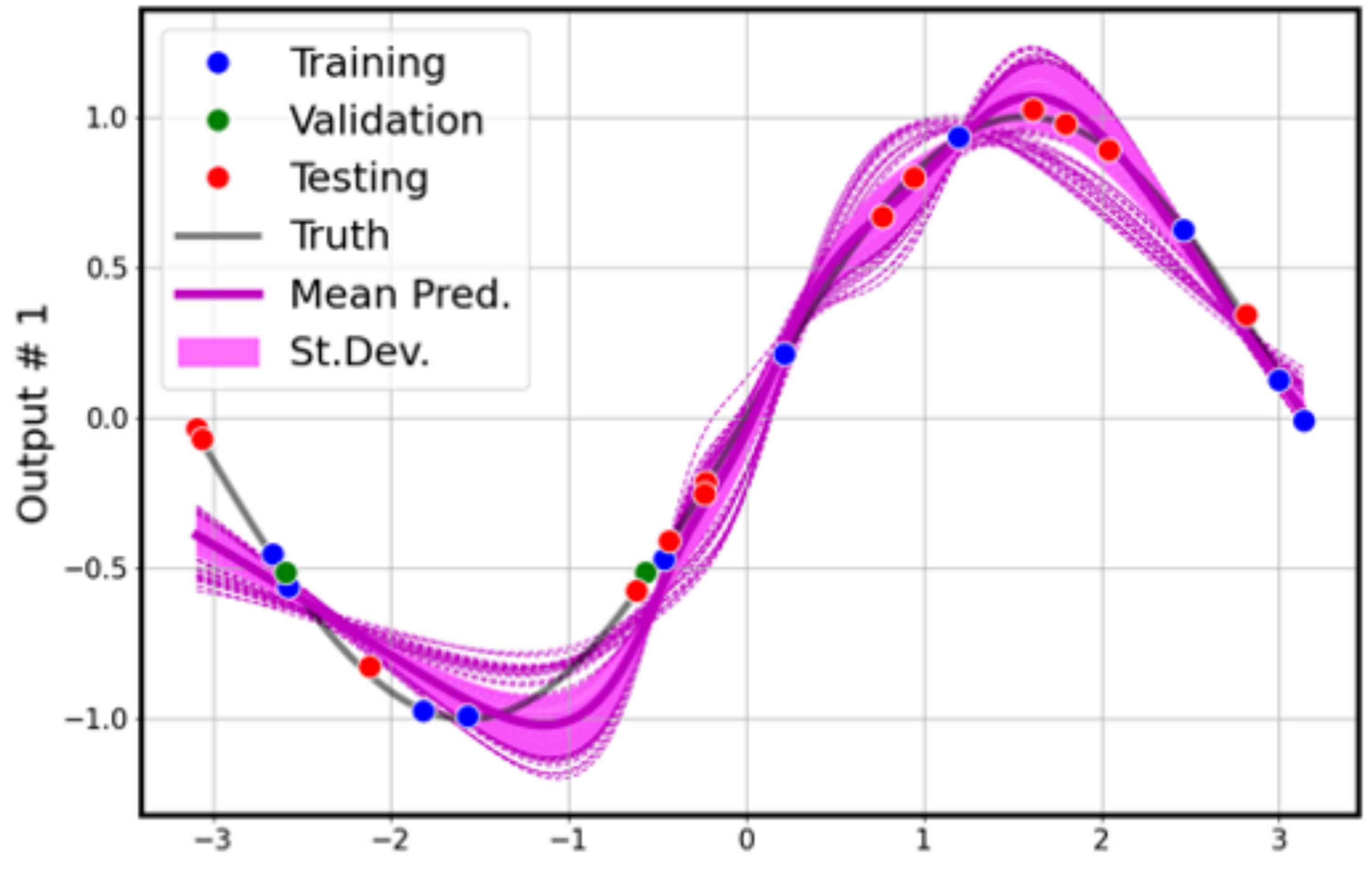
```
class VI_NN(QUiNNBase):
    def __init__(self, nnmodule, verbose=False):
        super(VI_NN, self).__init__(nnmodule)
        self.bmodel = BNet(nnmodule)
        self.verbose = verbose
```

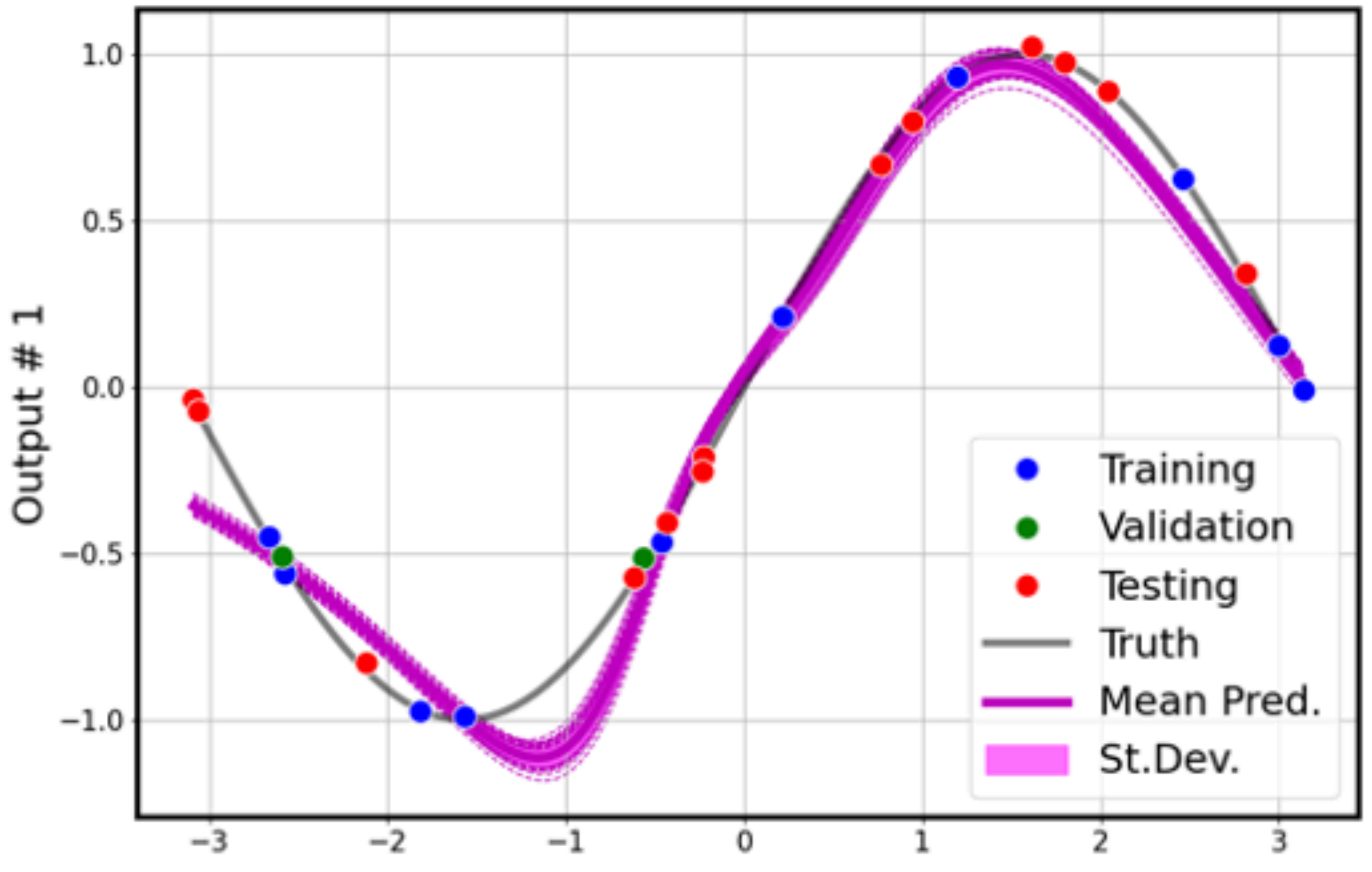
Variational Inference

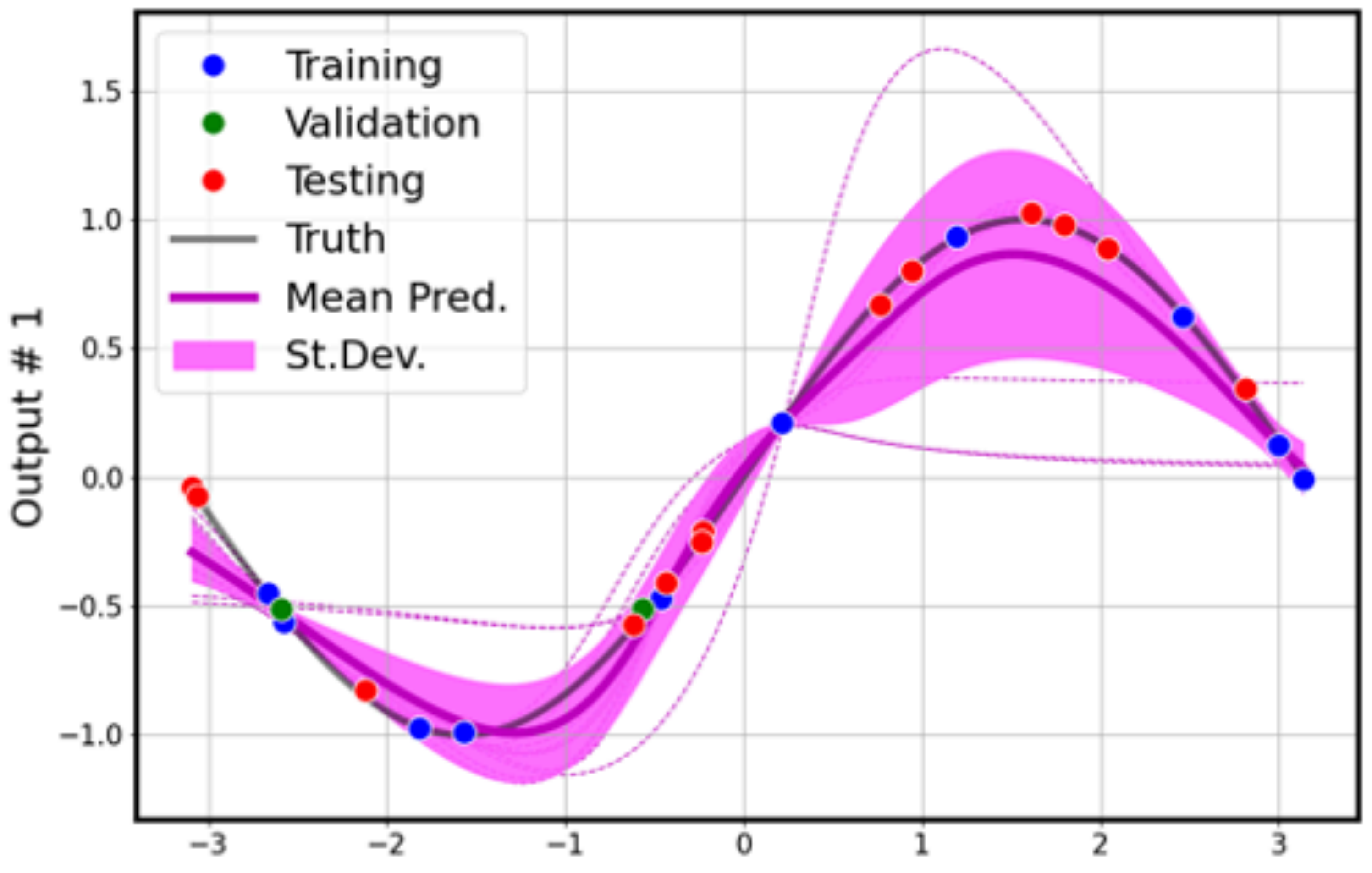
```
uqnet = Ens_NN(nnet, nens=nmc)
```

```
class Ens_NN(QUiNNBase):
    def __init__(self, nnmodule, nens=1, verbose=False):
        super(Ens_NN, self).__init__(nnmodule)
        self.verbose = verbose
        self.nens = nens
```

Ensembling

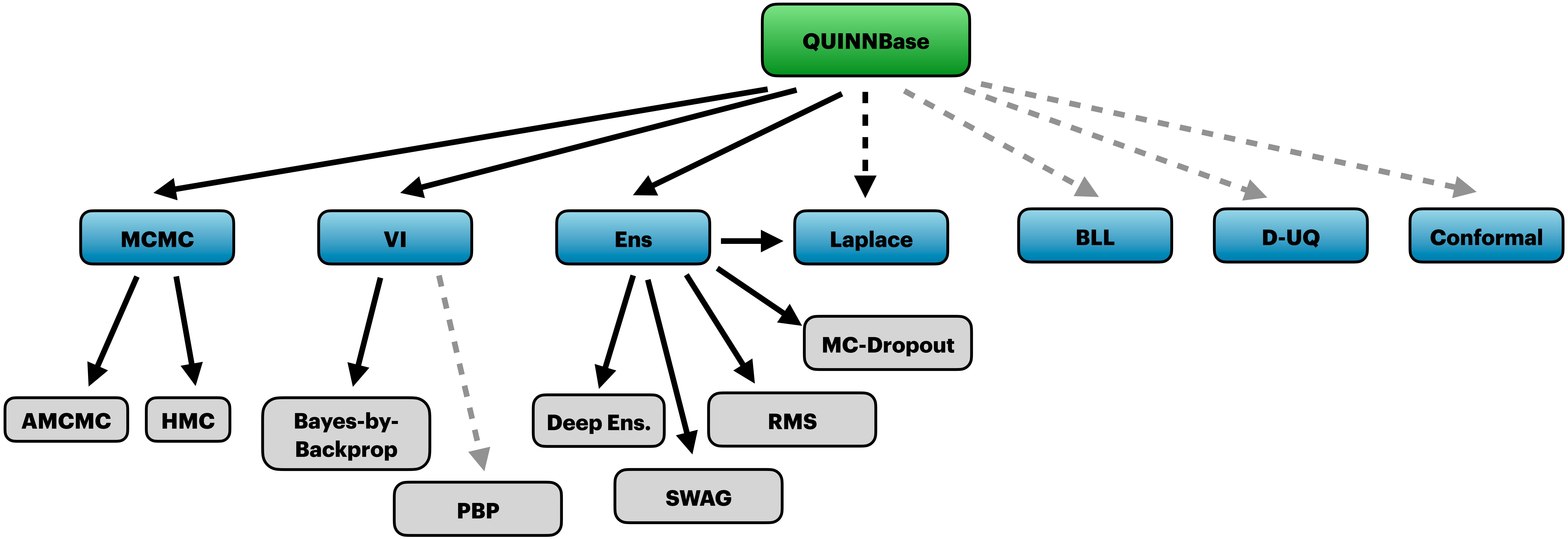




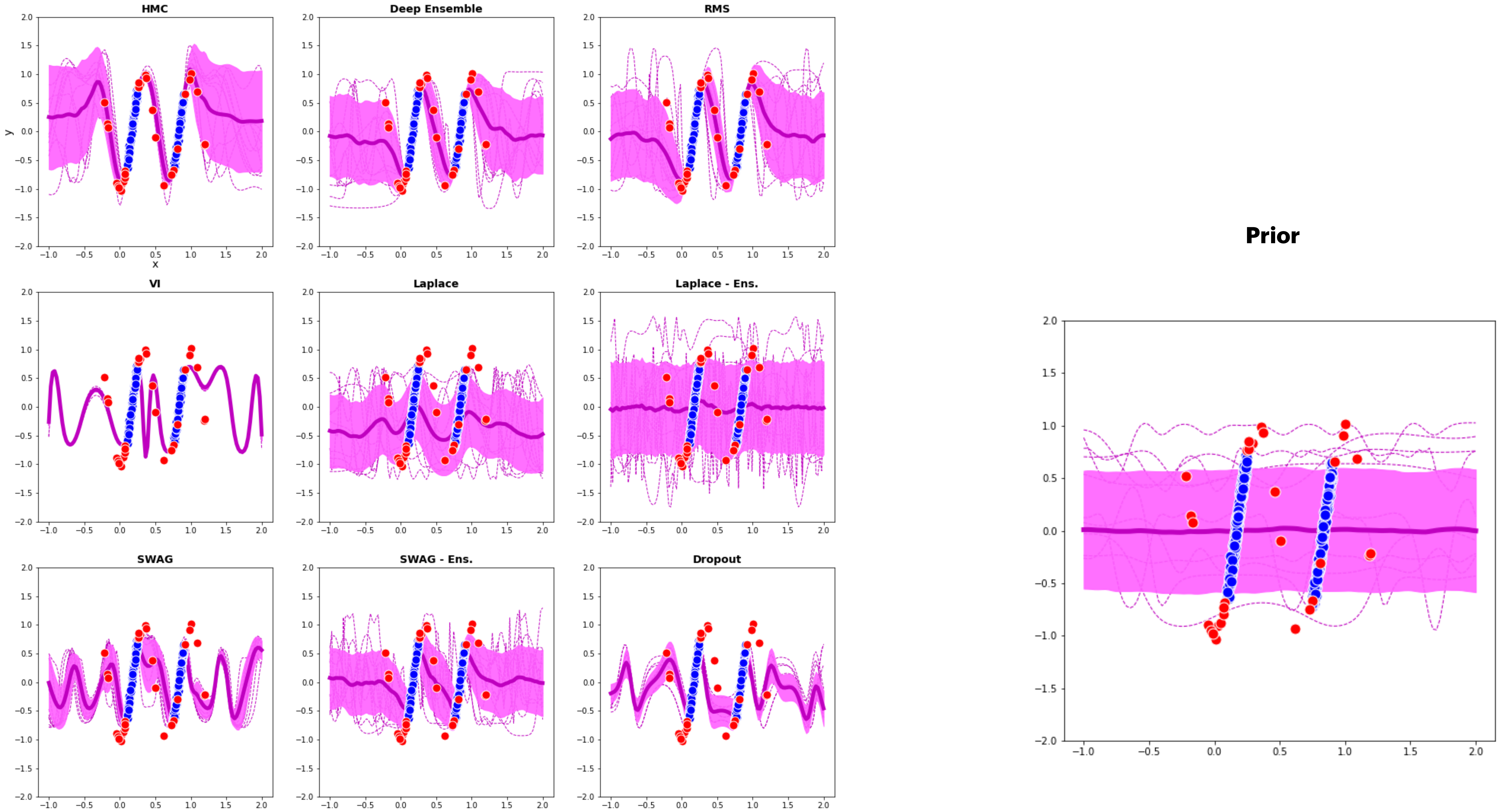


QUiNN: github.com/sandialabs/quinn

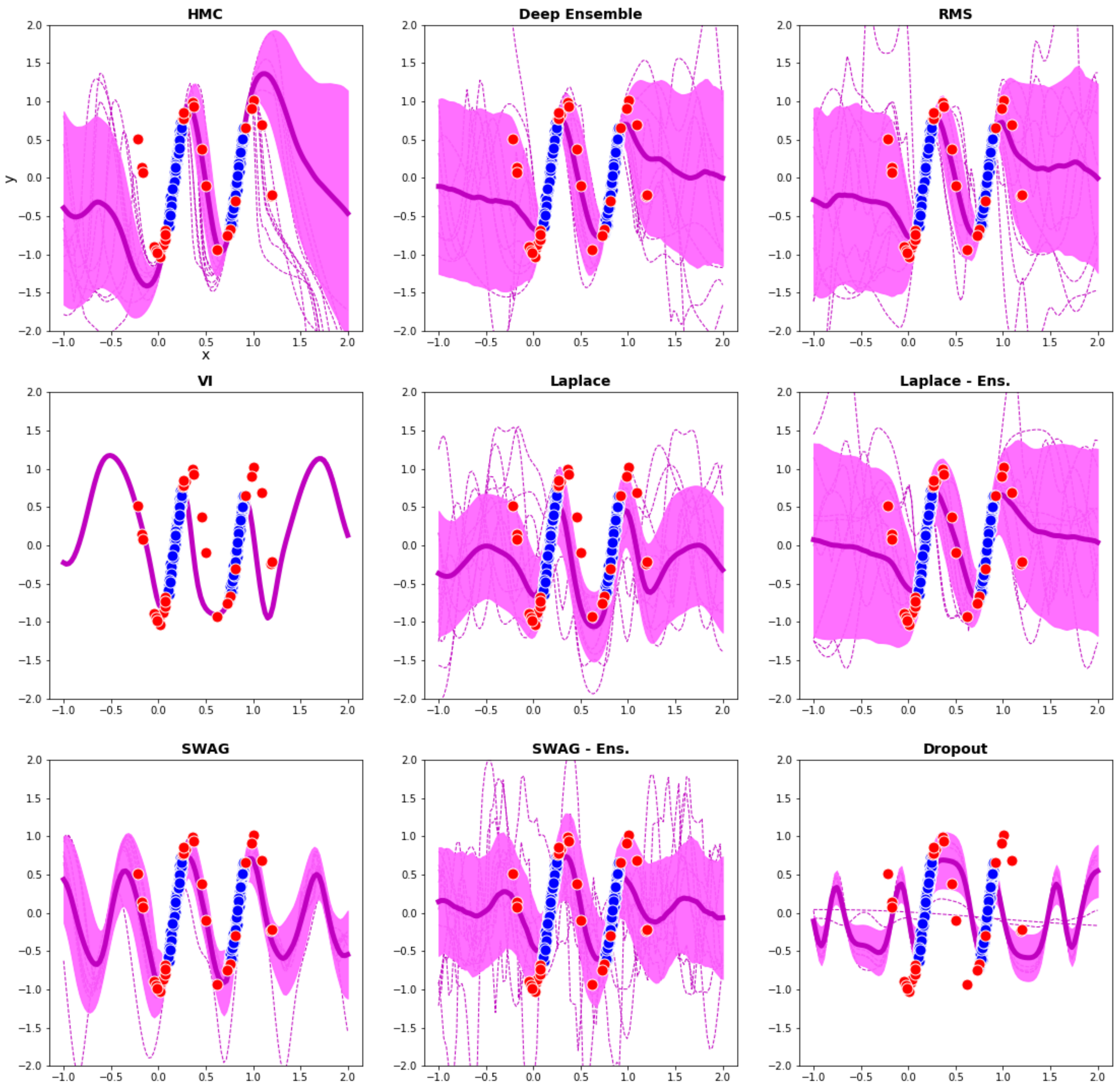
```
uqnet = QUINNBase(torch.nn.module)
```

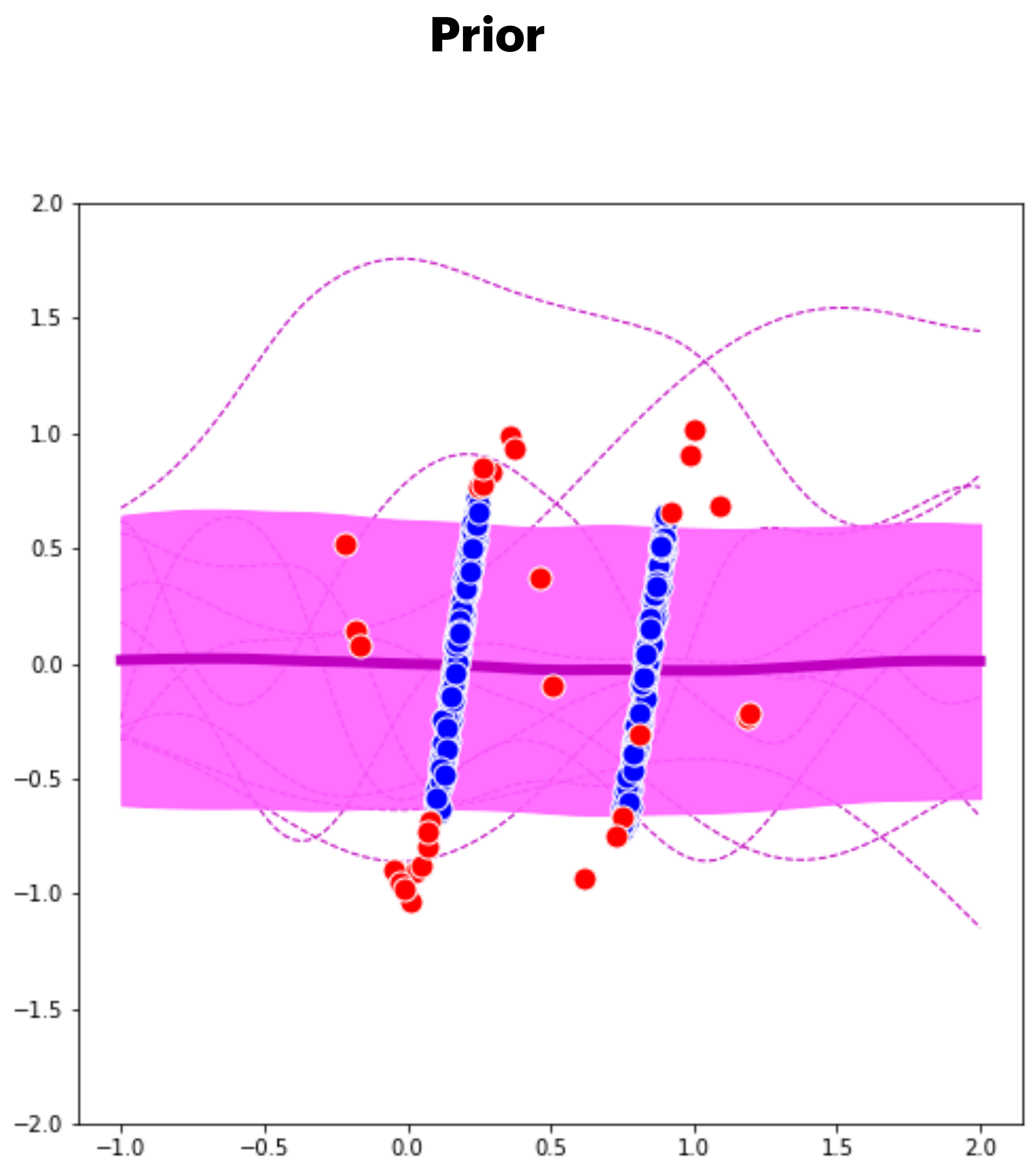


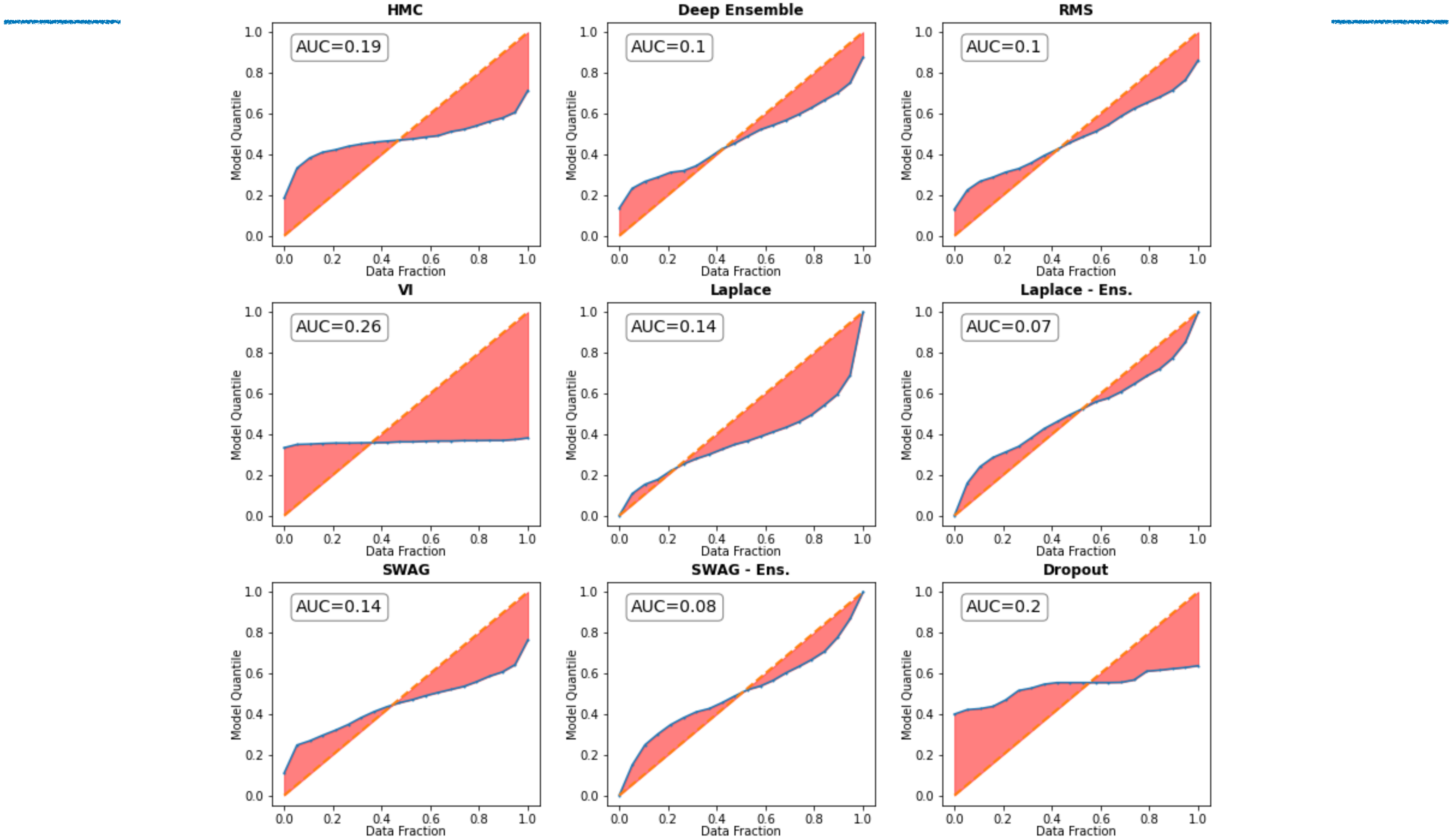
MLP


HMC
Deep Ensemble
RMS
VI
Laplace
Laplace - Ens.
SWAG
SWAG - Ens.
Dropout
Prior
x
y

WP-ResNet


HMC
Deep Ensemble
RMS
VI
Laplace
Laplace - Ens.
SWAG
SWAG - Ens.
Dropout
x
y


Prior


HMC
AUC=0.19
Deep Ensemble
AUC=0.1
RMS
AUC=0.1
VI
AUC=0.26
Laplace
AUC=0.14
Laplace - Ens.
AUC=0.07
SWAG
AUC=0.14
SWAG - Ens.
AUC=0.08
Dropout
AUC=0.2
Model Quantile
Data Fraction
0.0
0.2
0.4
0.6
0.8
1.0

Summary

- UQ for NN
  - An attempt to overview the methods
  - Most methods rely on loss landscape
  - Metrics/diagnostics of accuracy
  - Major challenges

- ResNet/ODE:
  - Draw inspiration from ODE and infinite depth limit
  - ResNets regularize the learning problem, smoother loss/log-posterior surface
  - Weight parameterization (WP) allows regularization without losing much expressivity
  - Full Bayesian UQ treatment made more feasible with WP ResNets

- Implemented in QUiNN: github.com/sandialabs/quinn modular code as a wrapper to categories of methods (MCMC/HMC, VI, RMS, Ens, Laplace, Dropout)

Literature

General probabilistic NN:

- Z. Ghahramani, "Probabilistic machine learning and artificial intelligence". Nature 521, 452–459 (2015)
- D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks". Neural Computation 4 448–472 (1992)
- R. M. Neal, "Bayesian Learning for Neural Networks". Springer, New York (1996)

UQ for NN methods:

- D. Lévy, M. D. Hoffman, and J. Sohl-Dickstein, "Generalizing Hamiltonian Monte Carlo with Neural Networks". ICLR (2018)
- C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, "Weight uncertainty in neural networks". arXiv:1505.05424 (2015)
- J.M. Hernández-Lobato, R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks". ICML (2015)
- H. Ritter, A. Botev, D. Barber, "A Scalable Laplace Approximation for Neural Networks", ICLR (2018)
- E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, P. Hennig, "Laplace Redux-Effortless Bayesian Deep Learning" Advances in neural inf. proc. systems 34 (2021)
- Y. Gal, Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning". ICML (2016)
- B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles". NIPS'17. 6405–6416 (2017)
- T. Pearce, F. Leibfried, A. Brintrup, "Uncertainty in Neural Networks: Approximately Bayesian Ensembling". Artificial Intelligence and Statistics, 108:234-244 (2020)s" Machine Learning: Science and Technology, 3-4 (2022)
- Y. Yang , L. Hodgkinson, R. Theisen, J. Zou, J. E. Gonzalez , K. Ramchandran, M. Mahoney. "Taxonomizing local versus global structure in neural network loss landscapes", NIPS (2021)
- R. Anirudh, J. J. Thiagarajan. "Delta-UQ: Accurate Uncertainty Quantification via Anchor Marginalization", arxiv.org/abs/2110.02197 (2021)
- R. Krishnan, O. Tickoo, "Improving model calibration with accuracy versus uncertainty optimization". arXiv:2012.07923 (2020)

Literature

UQ for NN methods, cont.:

- W.J. Maddox, et al. , "A simple baseline for Bayesian uncertainty in deep learning". NIPS (2019)
- T Garipov et al., “Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs”. NIPS (2018)
- S. Fort, H. Hu, B. Lakshminarayanan , Deep Ensembles: A Loss Landscape Perspective”, arxiv.org/abs/1912.02757, (2019)
- H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, "Visualizing the Loss Landscape of Neural Nets, NIPS (2018)
- Y. Hu, J. Musielewicz, Z. W. Ulissi and A. J. Medford, "Robust and scalable uncertainty estimation with conformal prediction for machine-learned interatomic potentials" Machine Learning: Science and Technology, 3-4 (2022)
- L. Guo, H. Wu, W. Zhou, Y. Wang, T. Zhou, "IB-UQ: Information bottleneck based uncertainty quantification for neural function regression and neural operator learning", https://arxiv.org/abs/2302.03271 (2023)
- J. Postels et al, “On the Practicality of Deterministic Epistemic Uncertainty”, ICLR (2022)
- J. Watson, J. A Lin, P. Klink, J. Pajarinen, J. Peters, “Latent Derivative Bayesian Last Layer Networks”, AISTATS (2021)
- S. Lahlou, M. Jain, H. Nekoei, V. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, Y. Bengio “DEUP: Direct Epistemic Uncertainty Prediction”, TMLR (2023)
- R. Egele, et al., "AutoDEUQ: Automated Deep Ensemble with Uncertainty Quantification," ICPR Proceedings, Montreal, QC, Canada, 2022 pp. 1908-1914 (2022)
- J-A. Goulet, L.-H. Nguyen, and S. Amiri, “Tractable approximate Gaussian inference for Bayesian neural networks”, JMLR, 20-1009, 22(251), pp. 1-23 (2021)

Neural ODE:

- R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, "Neural ordinary differential equations". NIPS'18 (2018).
- L. Ruthotto, E. Haber, "Deep neural networks motivated by partial differential equations". arXiv preprint arXiv:1804.04272 (2018)
- W. E, "A Proposal on Machine Learning via Dynamical Systems". Commun. Math. Stat. 5, 1–11 (2017)

Literature

Benchmarks:

- UCI Dataset, https://archive.ics.uci.edu/datasets
- J. Yao, W. Pan, S. Ghosh, F. Doshi-Velez, "Quality of Uncertainty Quantification for Bayesian Neural Network Inference”, https://arxiv.org/abs/1906.09686 (2019)
- J. Navratil, B. Elder, M. Arnold, S. Ghosh, P. Sattigeri, “Uncertainty Characteristics Curves: A Systematic Assessment of Prediction Intervals”, https://arxiv.org/abs/2106.00858 (2021)
- Z. Nado et al. “Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning”, https://arxiv.org/abs/2106.04015 (2021), https://github.com/google/uncertainty-baselines
- B. Staber, S. Da Veiga, “Benchmarking Bayesian neural networks and evaluation metrics for regression tasks”, https://arxiv.org/abs/2206.06779 (2022)
- L. Basora, A. Viens, M. Arias Chao, X. Olive, “A Benchmark on Uncertainty Quantification for Deep Learning Prognostics”, https://arxiv.org/abs/2302.04730 (2023)

Additional

Randomized MAP Sampling (RMS)

[Pearce, 2020]

- Consider log-posterior: $-\log P(w|y) = ||y - NN_w(x)||^2 + R(w)$

- Consider regularized training problem $\min \left(\alpha ||y - NN_w(x)||^2 + \beta ||w - w^*||^2 \right)$

- If one samples w^* from prior $\sim e^{-R(w)}$, the set of deterministic solutions <u>approximately</u> forms the posterior $P(w|y)$

- It is exact for gaussian priors, linear models:
but the authors show that it extends well to larger class, including NNs

- What is missing: proper attribution of uncertainty: is it really RMS or the initialization that drives the good results?