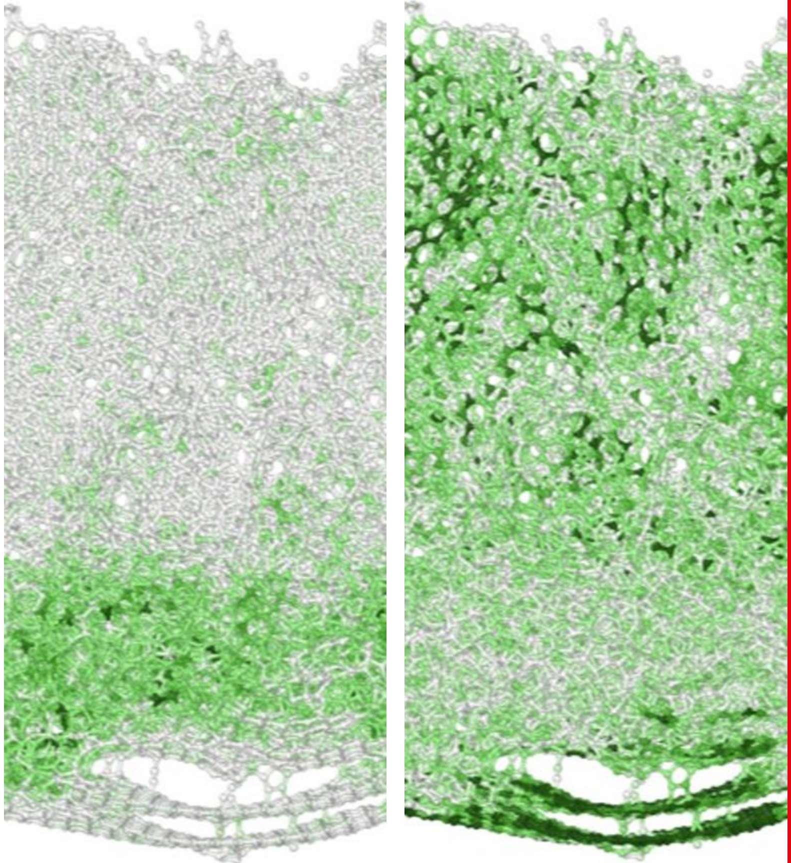


Trustworthy machine learning for materials

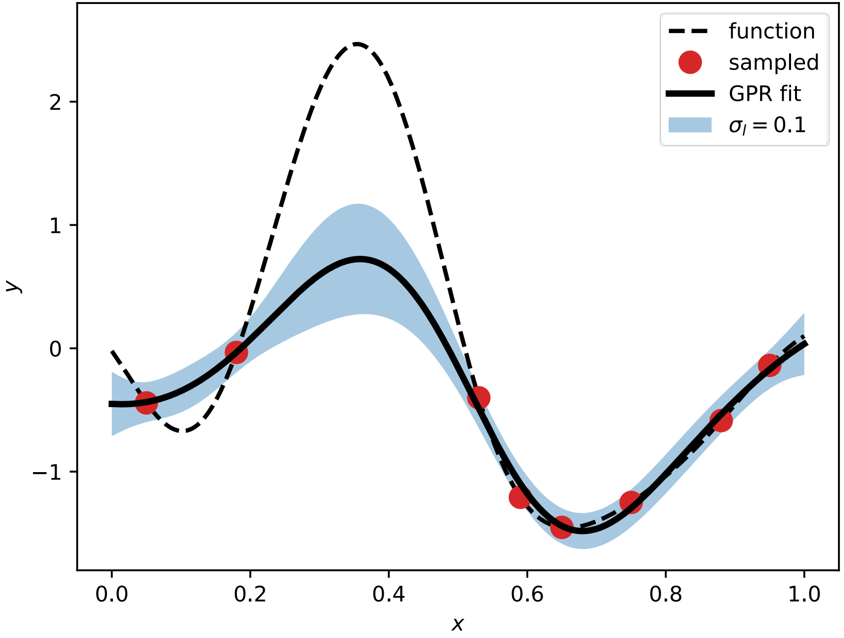
Federico Grasselli – COSMO Lab

EPFL

Uncertainty Quantification (UQ)

- Pillar of Scientific Method
- ML statistical nature
- Sources of uncertainty
- So far scarcely employed:
  - lack of standards
  - large cost (training and evaluation)
  - ad hoc training:

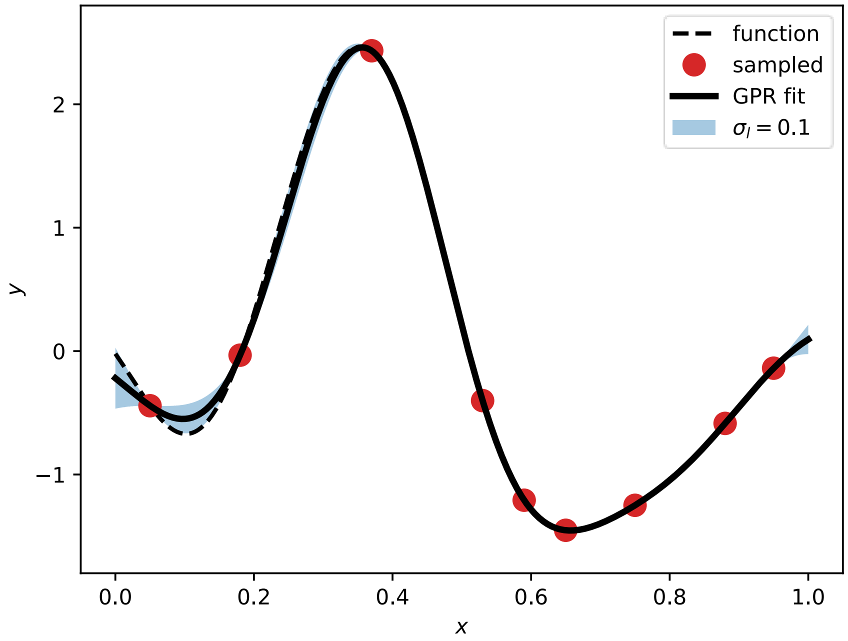
    (MC dropout, deep and shallow ensembles, Gaussian mixture models, committees)



Uncertainty Quantification (UQ)

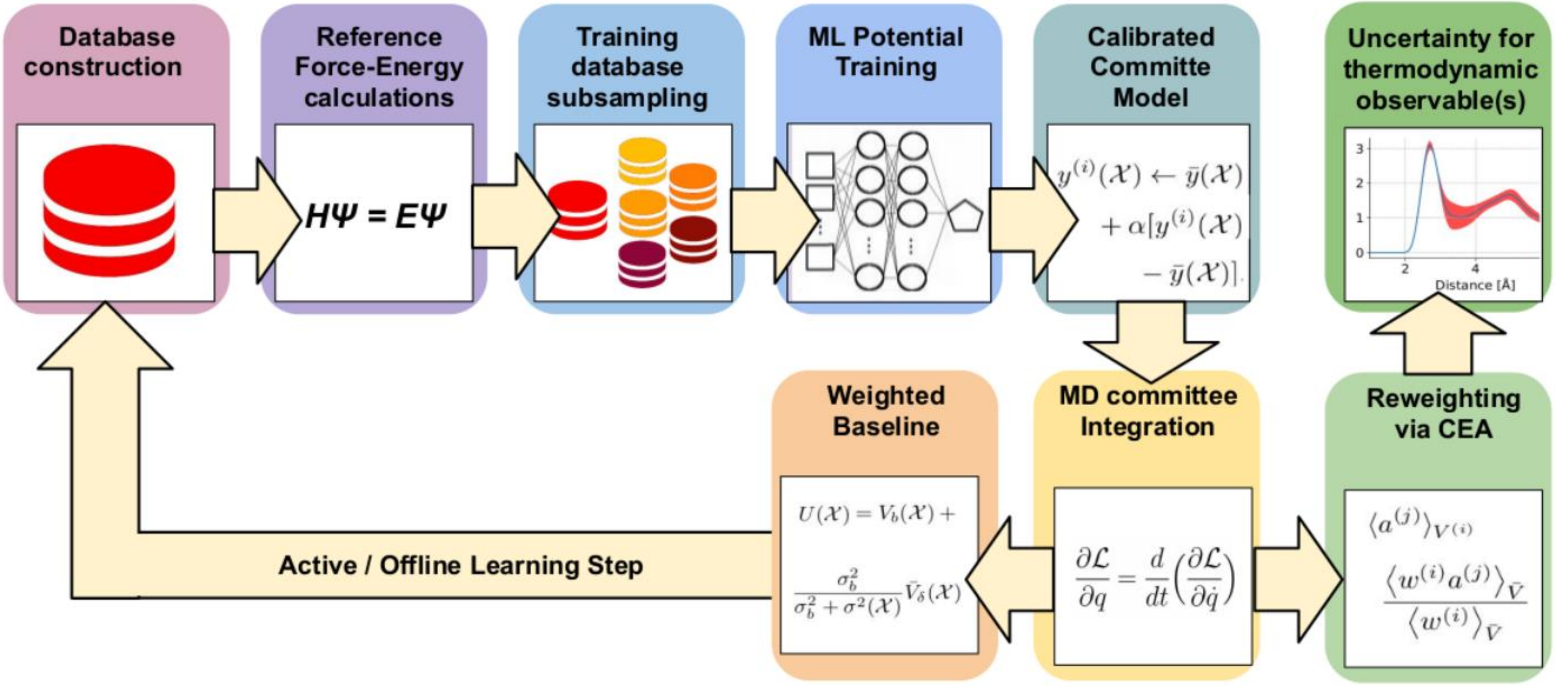
- Pillar of Scientific Method
- ML statistical nature
- Sources of uncertainty
- So far scarcely employed:
  - lack of standards
  - large cost (training and evaluation)
  - ad hoc training:

    (MC dropout, deep and shallow ensembles, Gaussian mixture models, committees)



Use of UQ in atomistic simulations

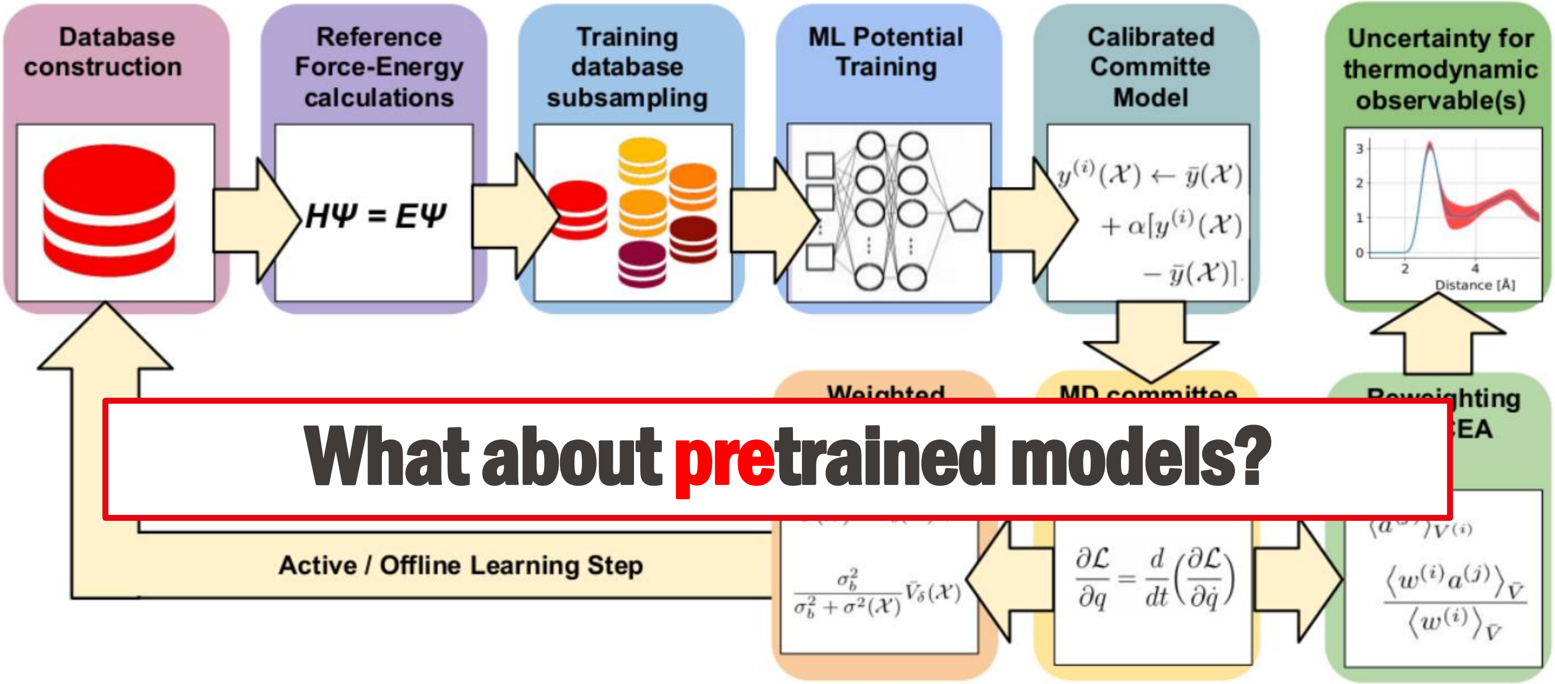
- Uncertainty aware simulations
- Reweight via committee of models



Imbalzano, Zhuang, Kapil, Rossi, Engel, Grasselli, and Ceriotti, *J. Chem. Phys.* **154**, 074102 (2021)

Use of UQ in atomistic simulations

- Uncertainty aware simulations
- Reweight via committee of models



What about pretrained models?

Imbalzano, Zhuang, Kapil, Rossi, Engel, Grasselli, and Ceriotti, *J. Chem. Phys.* **154**, 074102 (2021)

Prediction Rigidity (PR)

- Loss:

$$\mathcal{L}(\mathbf{w}|D) = \sum_{i=1}^{N_{\mathrm{train}}} \ell[\tilde{y}_i(\mathbf{x}_i, \mathbf{w}), y_i]$$

- How "rigid" is the prediction for a given input $\mathbf{x}_\star$?

Prediction Rigidity (PR)

- Loss:

$$\mathcal{L}(\mathbf{w}|D) = \sum_{i=1}^{N_{\text{train}}} \ell[\tilde{y}_i(\mathbf{x}_i, \mathbf{w}), y_i]$$

- How “rigid” is the prediction for a given input $\mathbf{x}_\star$?
- Constrained minimization of $\mathcal{L}_c = \mathcal{L} + \lambda\big(\epsilon_\star - \tilde{y}(\mathbf{x}_\star, \mathbf{w})\big)$

Prediction Rigidity (PR)

- Loss:

$$\mathcal{L}(\mathbf{w}|D) = \sum_{i=1}^{N_{\text{train}}} \ell[\tilde{y}_i(\mathbf{x}_i, \mathbf{w}), y_i]$$

- How "rigid" is the prediction for a given input $\mathbf{x}_\star$?
- Constrained minimization of $\mathcal{L}_c = \mathcal{L} + \lambda\big(\epsilon_\star - \tilde{y}(\mathbf{x}_\star, \mathbf{w})\big)$
- For $\epsilon_\star \approx \tilde{y}(\mathbf{x}_\star, \mathbf{w}_o)$ we have

$$\mathcal{L}_c(\epsilon_\star|D) \approx \mathcal{L}(\mathbf{w}_o|D) + \frac{1}{2} R_\star \big(\epsilon_\star - \tilde{y}(\mathbf{x}_\star, \mathbf{w}_o)\big)^2$$

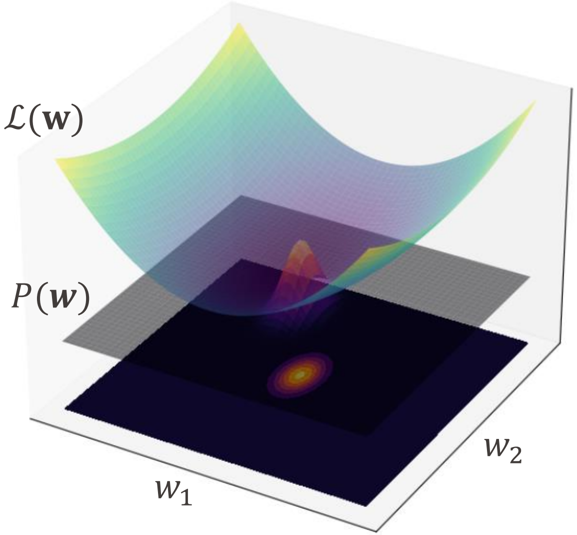
- with

$$R_\star \equiv \left.\frac{\partial^2 \mathcal{L}}{\partial \epsilon_\star^2}\right|_{\epsilon_\star = \tilde{y}(\mathbf{x}_\star, \mathbf{w}_o)} = \left[\left.\frac{\partial \tilde{y}_\star}{\partial \mathbf{w}}\right|_{\mathbf{w}_o} \left[\left.\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^2}\right|_{\mathbf{w}_o} \right]^{-1} \left.\frac{\partial \tilde{y}_\star}{\partial \mathbf{w}}\right|_{\mathbf{w}_o} \right]^{-1}$$

Bigi, Chong, Ceriotti & Grasselli, arXiv:2403.02251 (2024)

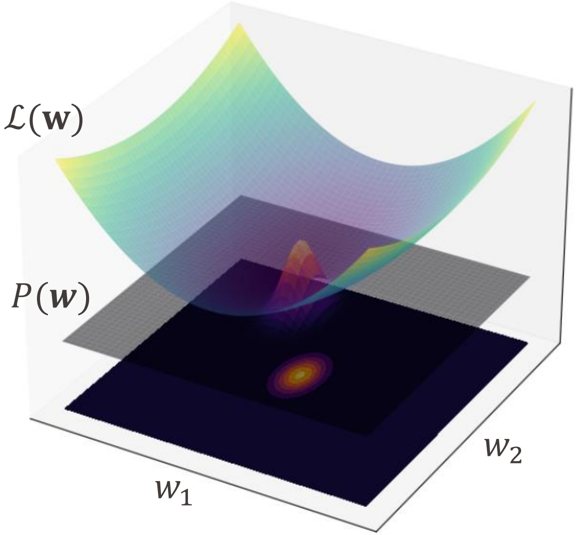
Prediction Rigidity (PR)

- Fitted models: $P(\mathbf{w}) \propto \exp(-\mathcal{L}(\mathbf{w}))$
- Laplace approximation:
  - 2^{nd}-order approximation of the loss
  - Gaussian approx. of the probability density



Prediction Rigidity (PR)

- Fitted models: $P(\mathbf{w}) \propto \exp(-\mathcal{L}(\mathbf{w}))$
- Laplace approximation:
  - 2nd-order approximation of the loss
  - Gaussian approx. of the probability density



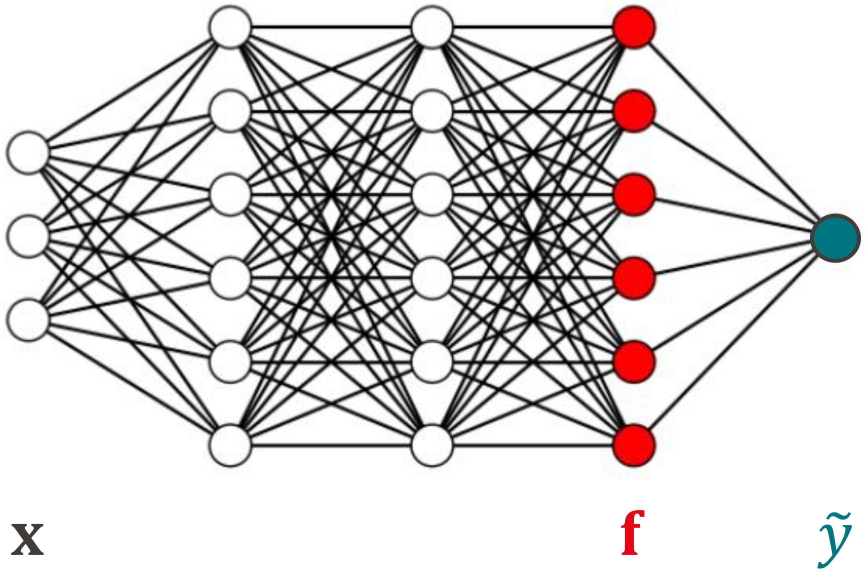
- Linear regression: $\tilde{y} = \mathbf{w} \cdot \mathbf{x} \rightarrow \frac{\partial \tilde{y}}{\partial \mathbf{w}} \equiv \mathbf{x}$
- PR is simple:

$$R_\star = [\mathbf{x}_\star \cdot \ [\mathbf{X}^\top \mathbf{X}]^{-1} \mathbf{x}_\star]^{-1}$$

- $\frac{1}{R_\star}$ has got the same shape of variance in Gaussian process regression

What about Neural Networks?

- Statistical theory of NNs. Training and over-parametrization
- Central Limit Theorem → infinitely wide NNs as Gaussian processes
- Last-layer features $\mathbf{f}$



[1] Lee et al. arXiv:1711.00165 (2017) [2] Jacot et al, NIPS (2018) [3] Lee et al, NIPS (2019)

Last-Layer Prediction Rigidity

- Uncertainty simplifies to (the inverse of) Last-Layer PR:

last-layer features for
the training set

last-layer features for
the prediction

predictive
uncertainty

$$\sigma_\star^2 = \alpha^2\, \mathbf{f}_\star^\top (\mathbf{F}^\top \mathbf{F} + \varsigma^2 \mathbf{I})^{-1} \mathbf{f}_\star$$

scale

regularization

Last-Layer Prediction Rigidity

- Uncertainty simplifies to (the inverse of) Last-Layer PR:

last-layer features for the training set

last-layer features for the prediction

predictive uncertainty

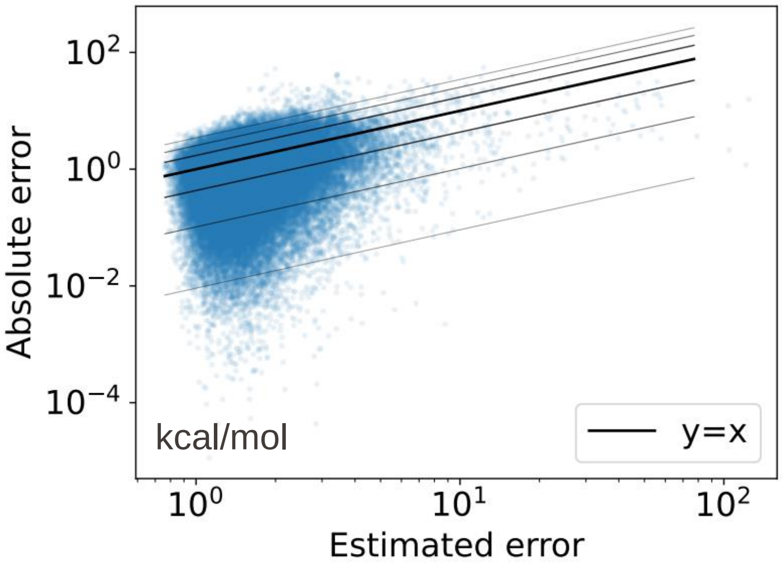
$$\sigma_\star^2 = \alpha^2\, \mathbf{f}_\star^\top (\mathbf{F}^\top \mathbf{F} + \varsigma^2 \mathbf{I})^{-1} \mathbf{f}_\star$$

scale

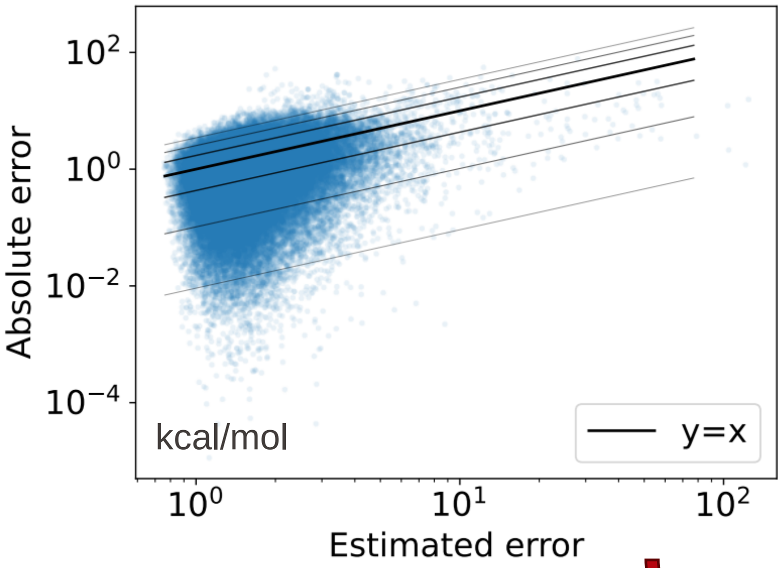
regularization

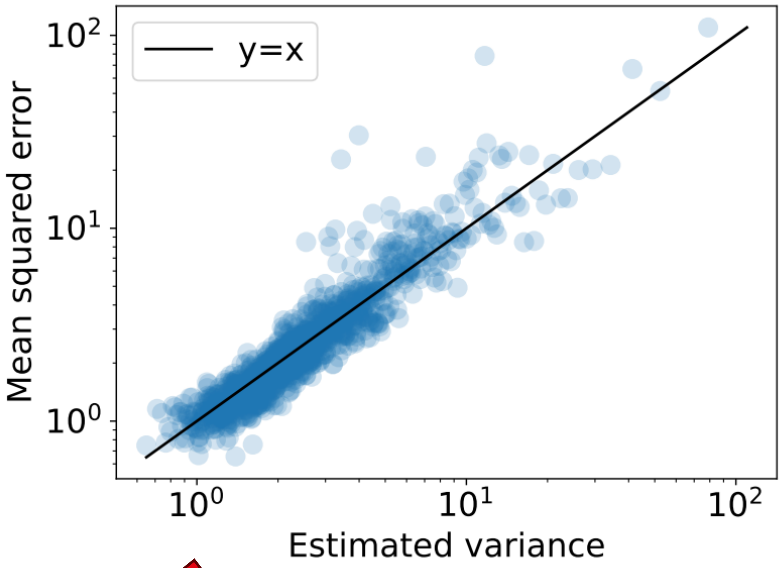
- Effectively the uncertainty of a linear model
- Very easy to calculate, two hyperparameters must be tuned
- Doesn't depend on target values of the training set

Last-Layer PR: results
QM9 dataset



Last-Layer PR: results QM9 dataset



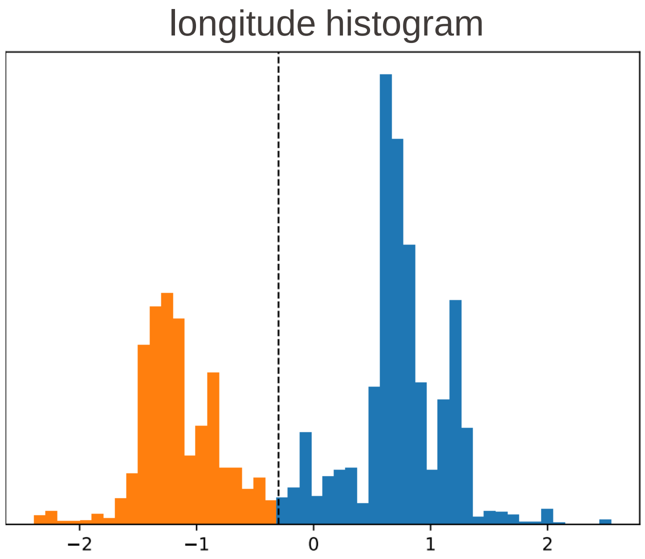


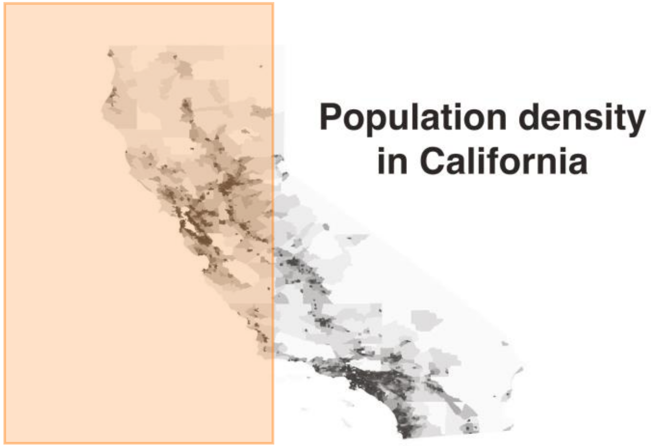
binning

Last-Layer PR results: California housing $

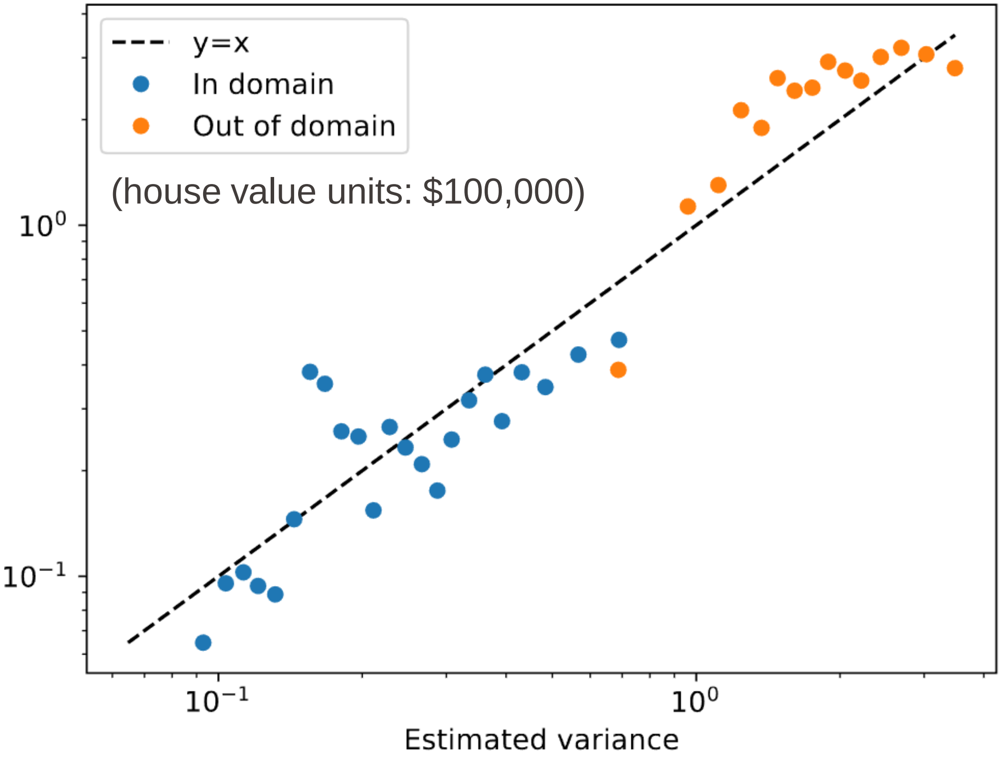
Population density in California

Last-Layer PR results: California housing $

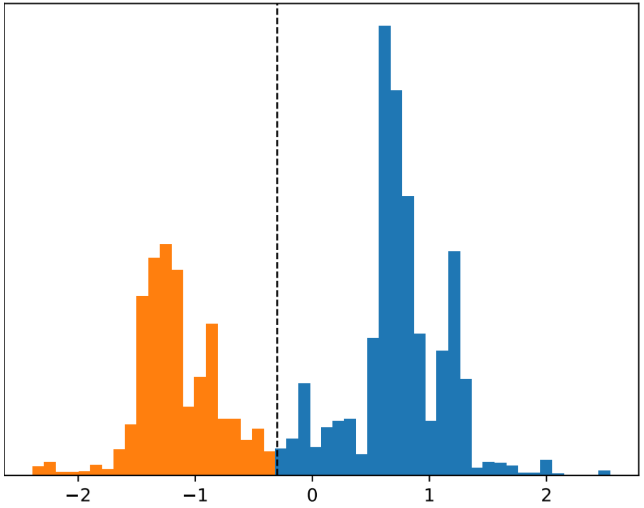


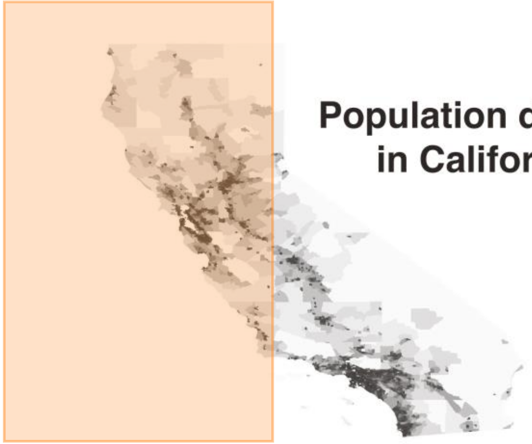


Last-Layer PR results: California housing $



longitude histogram



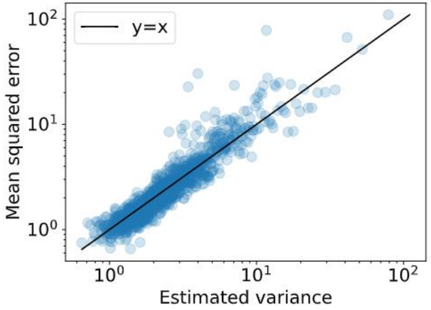


Population density in California

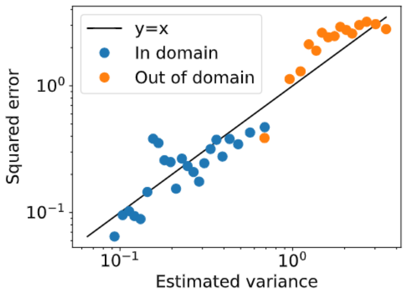
Federico Grasselli

Last-Layer PR results

- Very good uncertainty estimates across a wide variety of problems



Chemistry

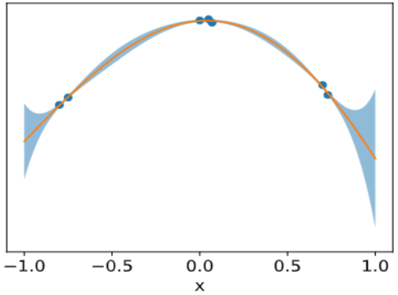


Housing prices

Last-Layer PR results

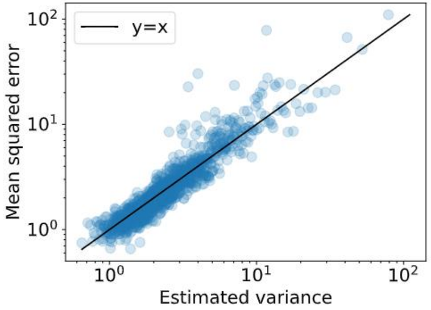
- Very good uncertainty estimates across a wide variety of problems

A toy model

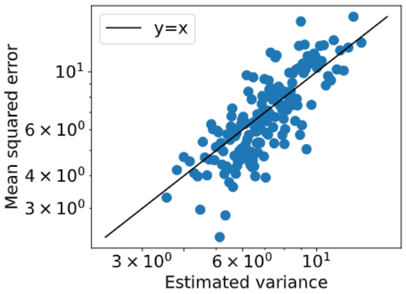


Dataset	RMSE				NLL			
	PBP	MCD	DE	LLPR	PBP	MCD	DE	LLPR
Concrete	5.67$_{\pm 0.09}$	**5.23**$_{\pm 0.12}$	6.03$_{\pm 0.13}$	**5.26**$_{\pm 0.25}$	3.16$_{\pm 0.02}$	**3.04**$_{\pm 0.02}$	**3.06**$_{\pm 0.04}$	**3.09**$_{\pm 0.07}$
Energy	1.80$_{\pm 0.05}$	1.66$_{\pm 0.04}$	2.09$_{\pm 0.06}$	**0.49**$_{\pm 0.03}$	2.04$_{\pm 0.02}$	1.99$_{\pm 0.02}$	1.38$_{\pm 0.05}$	**0.69**$_{\pm 0.07}$
Kin8nm	0.10$_{\pm 0.00}$	0.10$_{\pm 0.00}$	0.09$_{\pm 0.00}$	**0.08**$_{\pm 0.00}$	-0.90$_{\pm 0.01}$	-0.95$_{\pm 0.01}$	**-1.20**$_{\pm 0.00}$	-1.12$_{\pm 0.01}$
Naval	0.01$_{\pm 0.00}$	0.01$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	-3.73$_{\pm 0.01}$	-3.80$_{\pm 0.01}$	-5.63$_{\pm 0.01}$	**-7.07**$_{\pm 0.08}$
Power	4.12$_{\pm 0.03}$	**4.02**$_{\pm 0.04}$	4.11$_{\pm 0.04}$	**3.94**$_{\pm 0.07}$	2.84$_{\pm 0.01}$	**2.80**$_{\pm 0.01}$	**2.79**$_{\pm 0.01}$	2.83$_{\pm 0.02}$
Protein	4.73$_{\pm 0.01}$	4.36$_{\pm 0.02}$	4.71$_{\pm 0.03}$	**4.18**$_{\pm 0.02}$	2.97$_{\pm 0.00}$	2.89$_{\pm 0.00}$	**2.83**$_{\pm 0.01}$	2.91$_{\pm 0.01}$
Wine	**0.64**$_{\pm 0.01}$	**0.62**$_{\pm 0.01}$	**0.64**$_{\pm 0.01}$	**0.63**$_{\pm 0.02}$	0.97$_{\pm 0.01}$	**0.93**$_{\pm 0.01}$	**0.94**$_{\pm 0.03}$	1.02$_{\pm 0.03}$
Yacht	**1.02**$_{\pm 0.05}$	**1.11**$_{\pm 0.08}$	1.58$_{\pm 0.11}$	**1.19**$_{\pm 0.16}$	1.63$_{\pm 0.02}$	1.55$_{\pm 0.03}$	**1.18**$_{\pm 0.05}$	1.58$_{\pm 0.20}$
Year	8.88$_{\pm N/A}$	**8.86**$_{\pm N/A}$	8.89$_{\pm N/A}$	8.91$_{\pm N/A}$	3.60$_{\pm N/A}$	3.59$_{\pm N/A}$	**3.35**$_{\pm N/A}$	3.61$_{\pm N/A}$

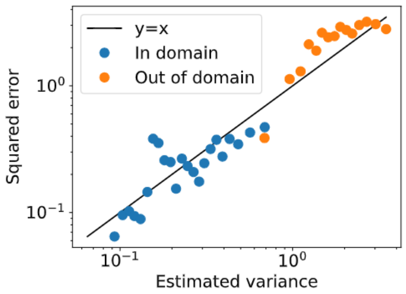
A standard benchmark



Chemistry



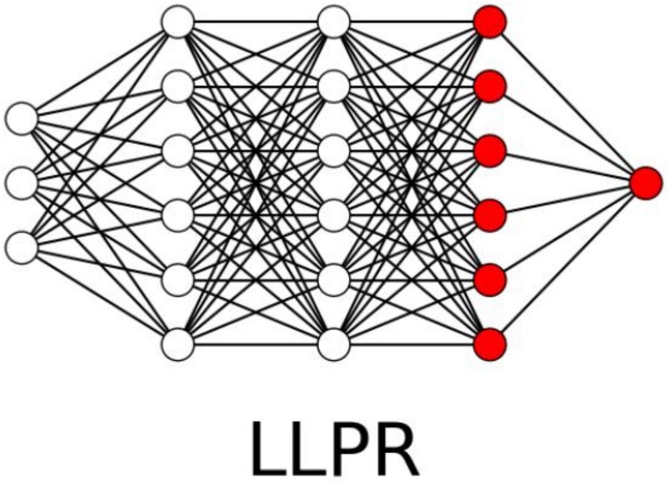
Weather prediction



Housing prices

Last-Layer PR: Summary

- Cheap, practical, scalable, a-posteriori
- Explains the success of last-layer approximations
- Pre-print on arxiv[1], code available @ https://github.com/frostedoyster/llpr



1 Bigi, Chong, Ceriotti & Grasselli, arXiv:2403.02251 (2024)

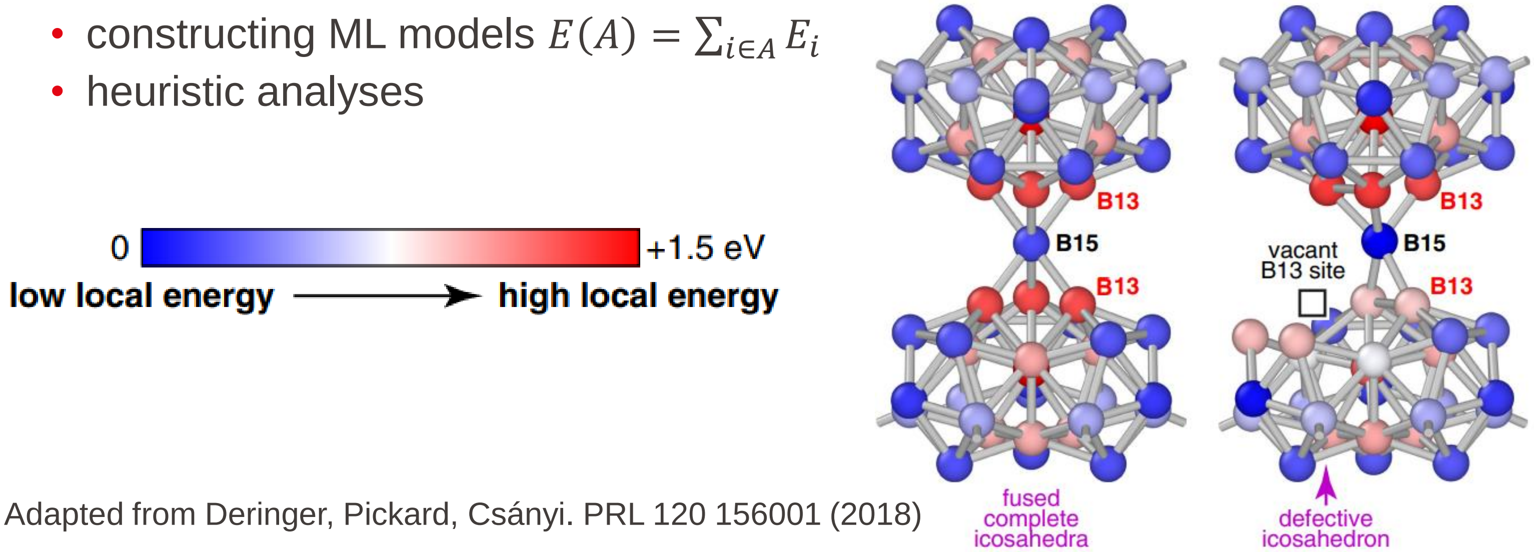
PR for local predictions

Federico Grasselli

- Atomistic models: local energies are **not** observables
- Yet used in
  - constructing ML models $E(A) = \sum_{i \in A} E_i$
  - heuristic analyses

PR for local predictions

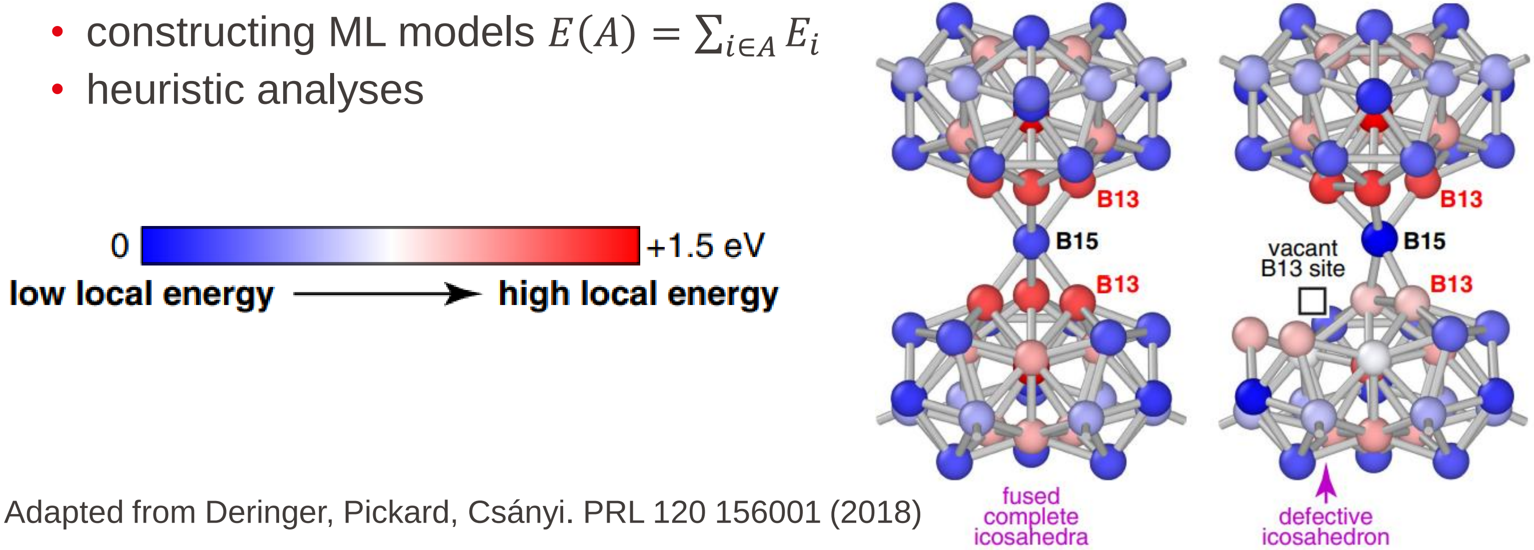
- Atomistic models: local energies are **not** observables
- Yet used in
  - constructing ML models $E(A) = \sum_{i \in A} E_i$
  - heuristic analyses



Adapted from Deringer, Pickard, Csányi. PRL 120 156001 (2018)

PR for local predictions

- Atomistic models: local energies are **not** observables
- Yet used in
  - constructing ML models $E(A) = \sum_{i\in A} E_i$
  - heuristic analyses

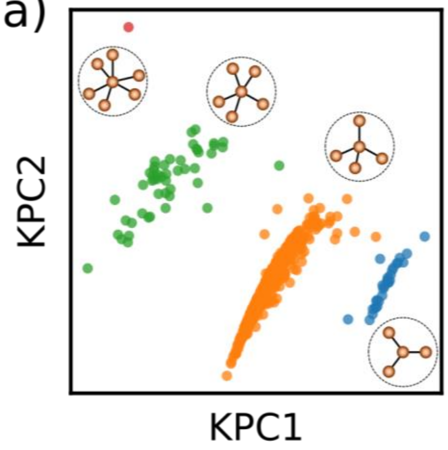


Adapted from Deringer, Pickard, Csányi. PRL 120 156001 (2018)

- How robust are these? Use the same formulation of constrained loss, but now for *local predictions*

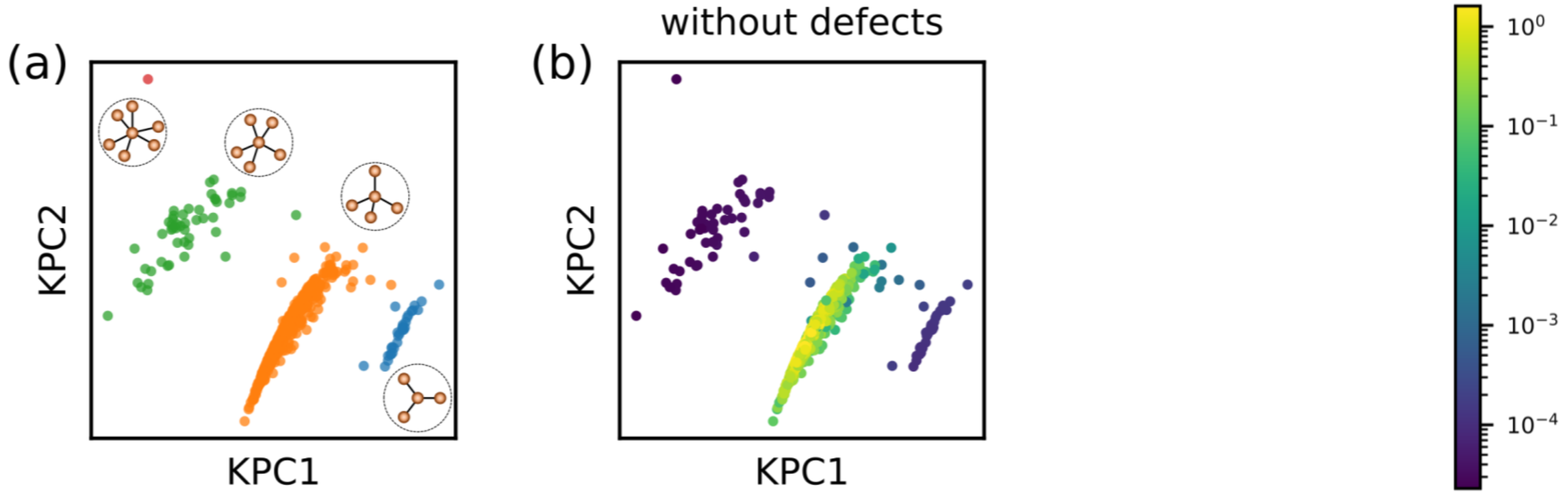
Local PR: amorphous silicon

(a)



Chong, Grasselli, Ben Mahmoud, Morrow, Deringer and Ceriotti, JCTC 19, 8020 (2023)

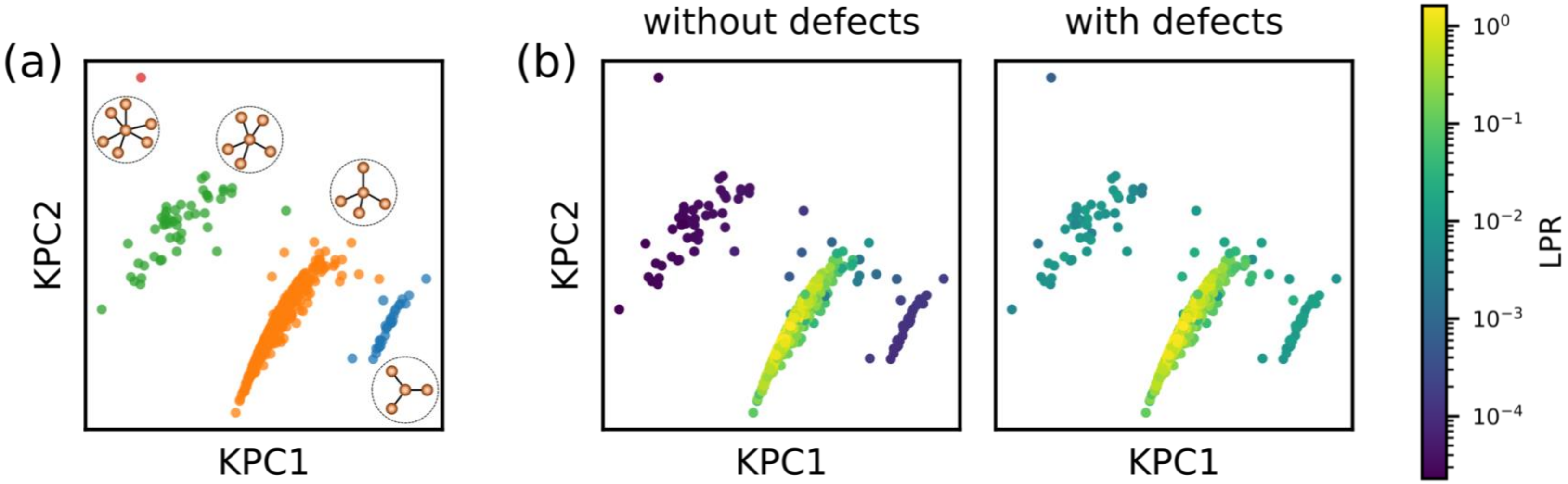
Local PR: amorphous silicon



Chong, Grasselli, Ben Mahmoud, Morrow, Deringer and Ceriotti, JCTC 19, 8020 (2023)

Local PR: amorphous silicon

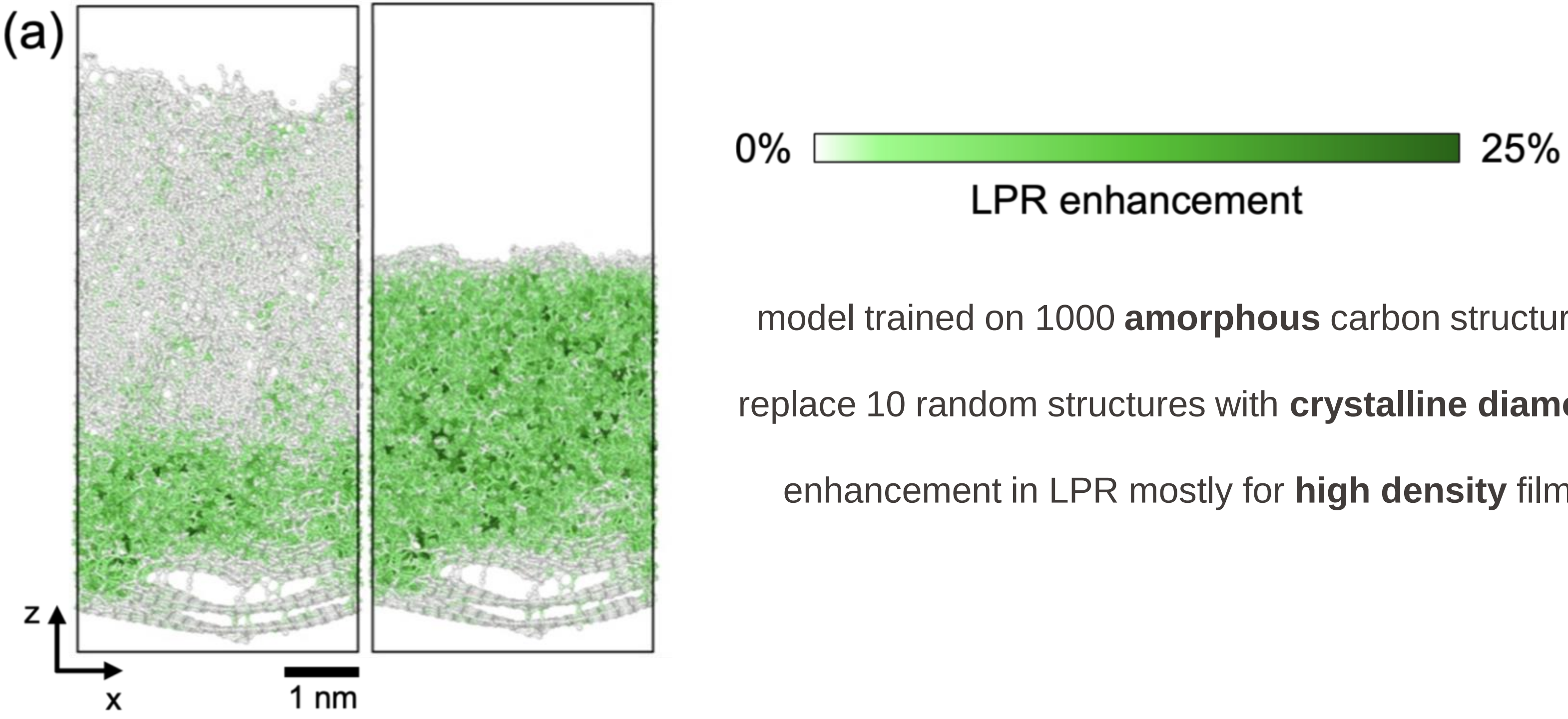
- Dramatic increase of local PR by adding structures containing under/over coordinated environments



Chong, Grasselli, Ben Mahmoud, Morrow, Deringer and Ceriotti, JCTC 19, 8020 (2023)

Local PR: carbon films

- Selective increase of local PR in low/high density carbon films



model trained on 1000 **amorphous** carbon structures

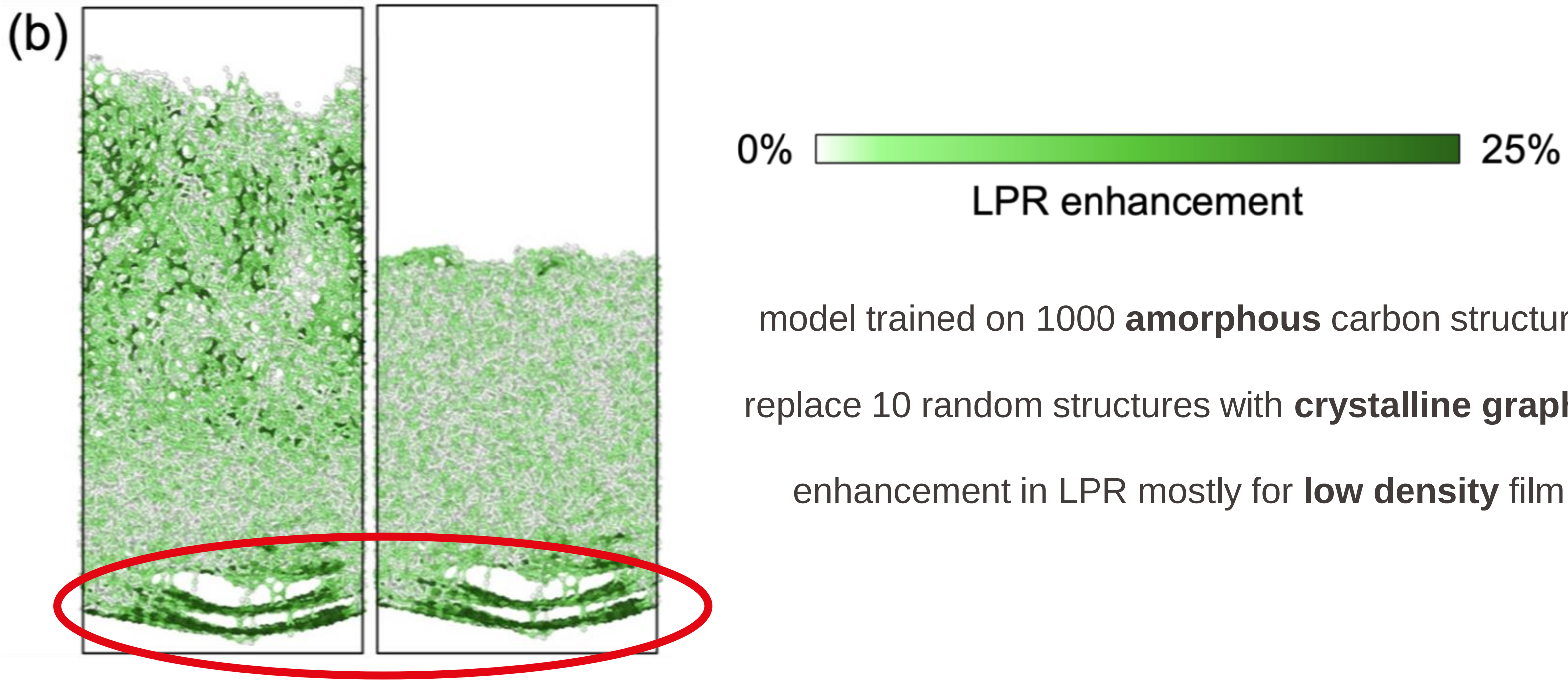
replace 10 random structures with **crystalline diamond**

enhancement in LPR mostly for **high density** film

Chong, Grasselli, Ben Mahmoud, Morrow, Deringer and Ceriotti, JCTC 19, 8020 (2023)

Local PR: carbon films

- Selective increase of local PR in low/high density carbon films



model trained on 1000 **amorphous** carbon structures

replace 10 random structures with **crystalline graphite**

enhancement in LPR mostly for **low density** film

Chong, Grasselli, Ben Mahmoud, Morrow, Deringer and Ceriotti, JCTC 19, 8020 (2023)

Remarks, Conclusions and Outlook

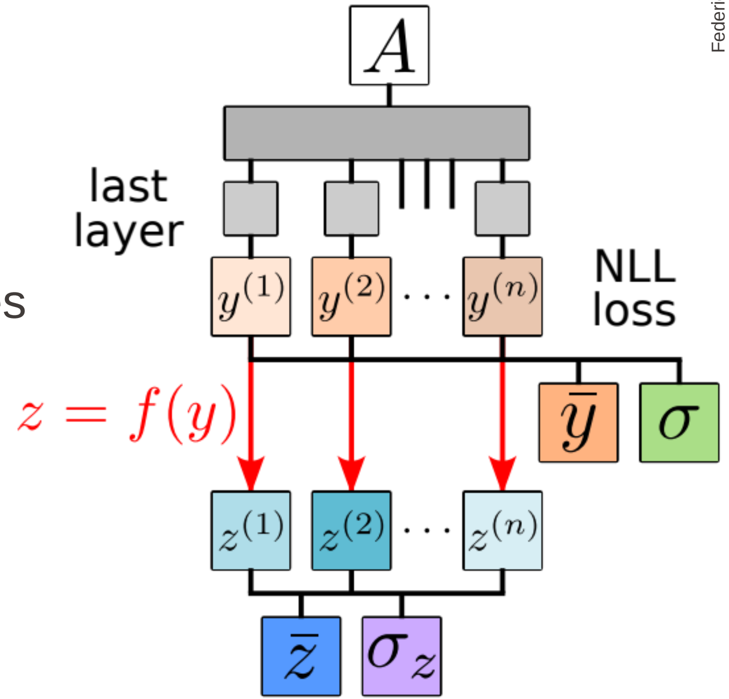
- Low-cost uncertainty on new predictions for pretrained models

- Constrained loss minimization is just a theoretical tool:
  - no need to train a model with constrained loss to get PR and uncertainties
  - no need for target values

- Rigidity of local predictions is readily obtained with same formalism

Remarks, Conclusions and Outlook

- Sample last-layer weights according to Laplace approximation → ensemble
- Propagate uncertainty to derived quantities
- Use it on thermodynamic observables



Kellner & Ceriotti, https://arxiv.org/abs/2402.16621 (2024)

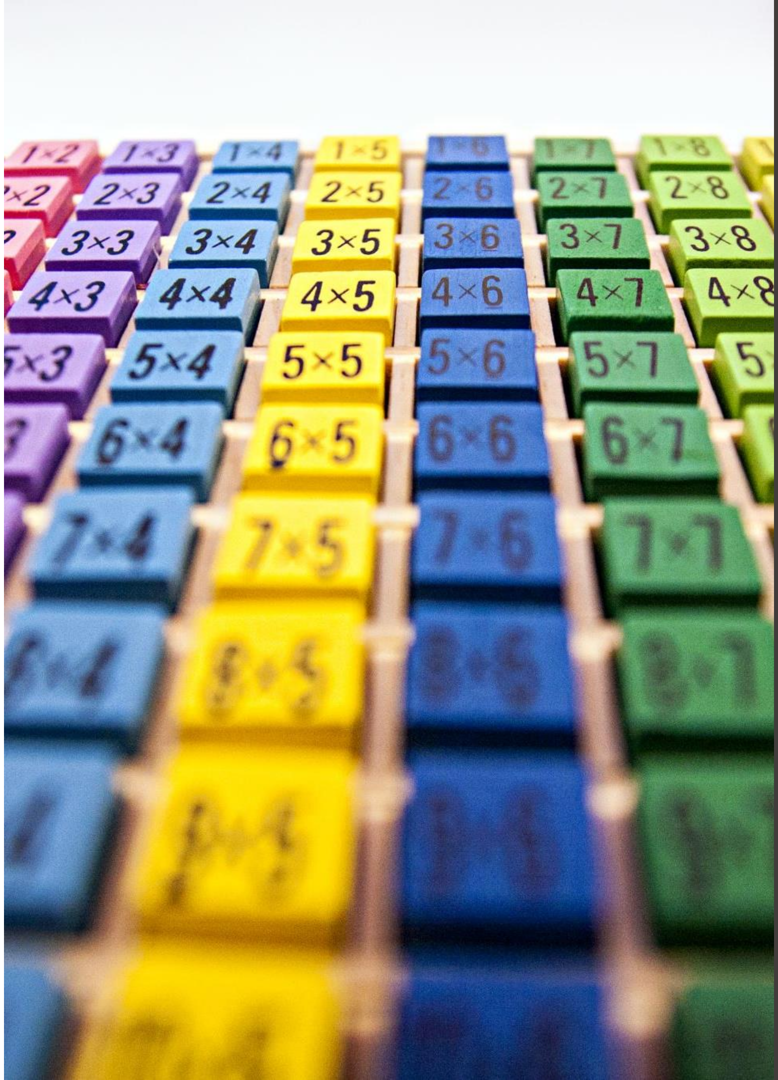
Acknowledgements

Sanggyu Chong, EPFL

Filippo Bigi, EPFL

Chiheb Ben Mahmoud, EPFL,
now Oxford University

Michele Ceriotti, EPFL



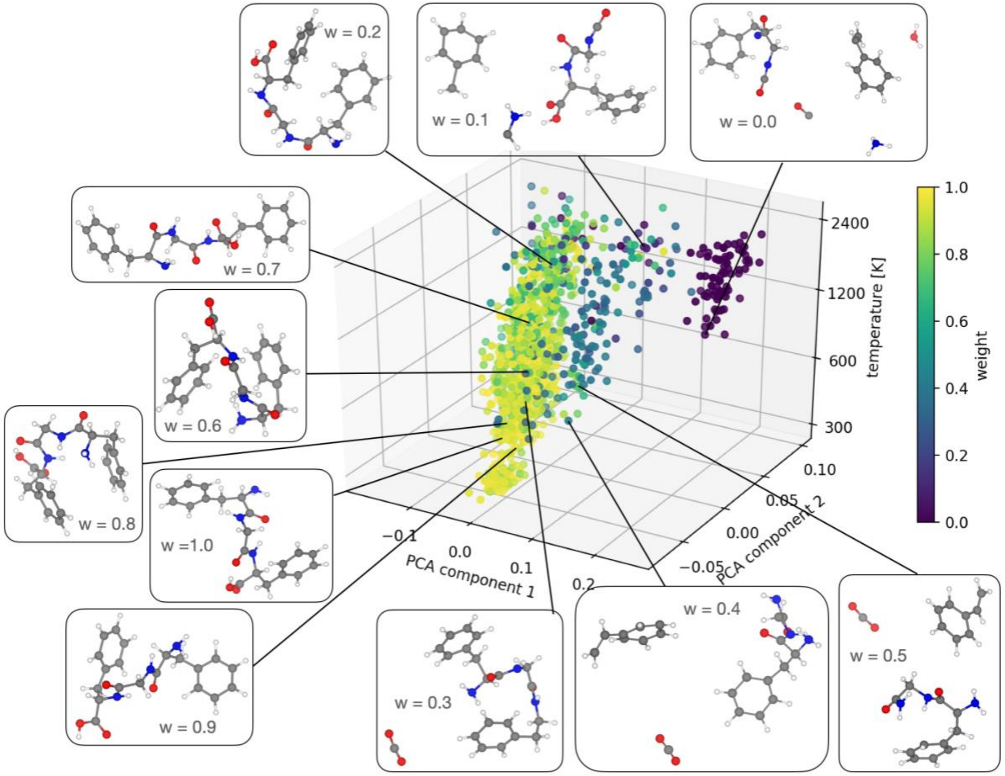
Backup slides

Federico Grasselli – COSMO Lab

EPFL

Use of UQ in atomistic simulations

- Uncertainty aware simulations



Imbalzano, Zhuang, Kapil, Rossi, Engel, Grasselli, and Ceriotti, *J. Chem. Phys.* **154**, 074102 (2021)

Extension to neural networks

- Statistical theory of NNs. Training and over-parametrization
- Central Limit Theorem → infinitely wide NNs as Gaussian processes
- Two deterministic kernels
  - Neural-Network Gaussian Process (NNGP)[1]: initialization
  - Neural Tangent Kernel (NTK)[2]: training
- Distribution of predictions[3] during training is Gaussian
- Evolution of mean and variance is deterministic!

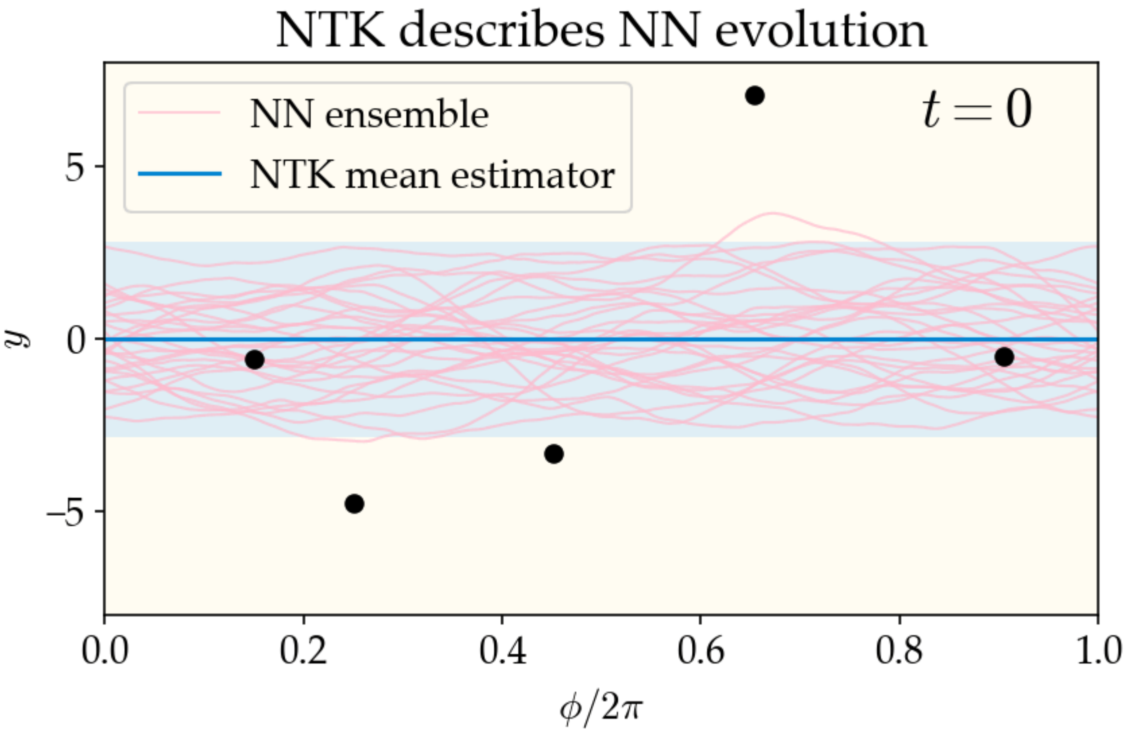
$$\mu_\star = \mathbf{k}_{\text{NTK}}(\star, \mathcal{D})\,\mathbf{K}_{\text{NTK}}^{-1}\left(\mathbf{I} - e^{-\eta \mathbf{K}_{\text{NTK}} t}\right)\mathbf{y}$$

$$\begin{aligned}\sigma_\star^2 = {} & k_{\text{NNGP}}(\star, \star) \\ & + \mathbf{k}_{\text{NTK}}(\star, \mathcal{D})\,\mathbf{K}_{\text{NTK}}^{-1}\left(\mathbf{I} - e^{-\eta \mathbf{K}_{\text{NTK}} t}\right)\mathbf{K}_{\text{NNGP}}\left(\mathbf{I} - e^{-\eta \mathbf{K}_{\text{NTK}} t}\right)\mathbf{K}_{\text{NTK}}^{-1}\,\mathbf{k}_{\text{NTK}}(\mathcal{D}, \star) \\ & - \mathbf{k}_{\text{NTK}}(\star, \mathcal{D})\,\mathbf{K}_{\text{NTK}}^{-1}\left(\mathbf{I} - e^{-\eta \mathbf{K}_{\text{NTK}} t}\right)\mathbf{k}_{\text{NNGP}}(\mathcal{D}, \star) \\ & - \mathbf{k}_{\text{NNGP}}(\star, \mathcal{D})\left(\mathbf{I} - e^{-\eta \mathbf{K}_{\text{NTK}} t}\right)\mathbf{K}_{\text{NTK}}^{-1}\,\mathbf{k}_{\text{NTK}}(\mathcal{D}, \star)\end{aligned}$$

[1] Lee et al. arXiv:1711.00165 (2017)
[2] Jacot et al, NIPS (2018)
[3] Lee et al, NIPS (2019)

Extension to neural networks

- Toy example (from Wikipedia)



Last-layer Prediction Rigidity (LLPR)

- A last-layer approximation of the prediction rigidity recovers:
  - the NNGP exactly at initialization

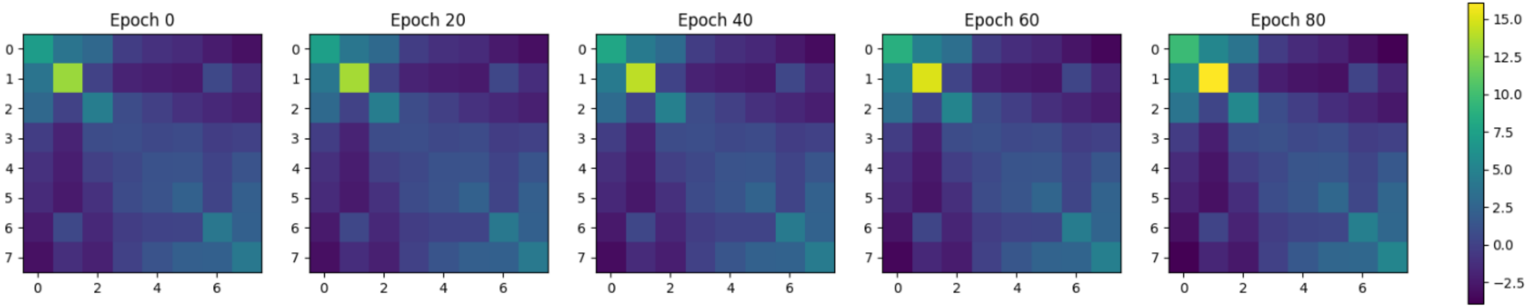
$$K_{\text{NNGP}}(\mathbf{x}_i, \mathbf{x}_j) \approx \sigma_w^2 \, \mathbf{f}_i^\top \mathbf{f}_j \Longrightarrow \mathbf{K}_{\text{NNGP}}(\mathcal{D}, \mathcal{D}) \approx \sigma_w^2 \, \mathbf{F}\mathbf{F}^\top$$

  - the NTK to a good approximation

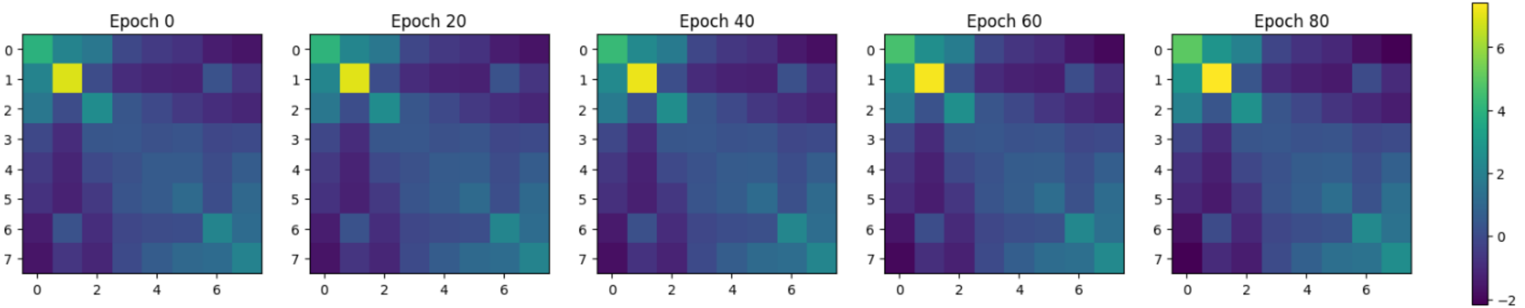
$$K_{\text{NTK}}(\mathbf{x}_i, \mathbf{x}_j) \approx c \left(\frac{\partial \tilde{y}(\mathbf{x}_i, \mathbf{w})}{\partial \mathbf{w}_L} \right)^\top \frac{\partial \tilde{y}(\mathbf{x}_j, \mathbf{w})}{\partial \mathbf{w}_L} = c \, \mathbf{f}_i^\top \mathbf{f}_j \Longrightarrow \mathbf{K}_{\text{NTK}}(\mathcal{D}, \mathcal{D}) \approx c \, \mathbf{F}\mathbf{F}^\top$$

Last-layer Prediction Rigidity (LLPR)

- full NTK



- Last-layer NTK: weaker dependence on “time”



California housing $: NN width test