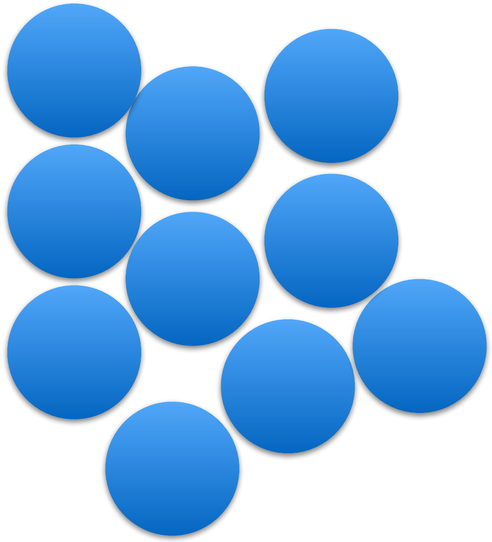
MAX PLANCK INSTITUTE
FOR POLYMER RESEARCH

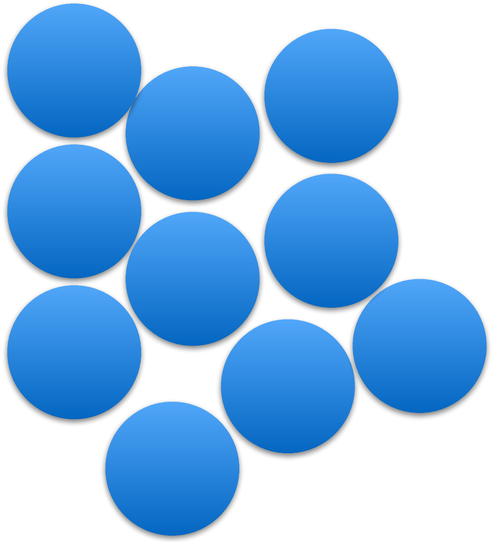
Conservation laws with ML for energy landscapes

Tristan Bereau
Max Planck Institute for Polymer Research
Mainz, Germany

Molecular dynamics

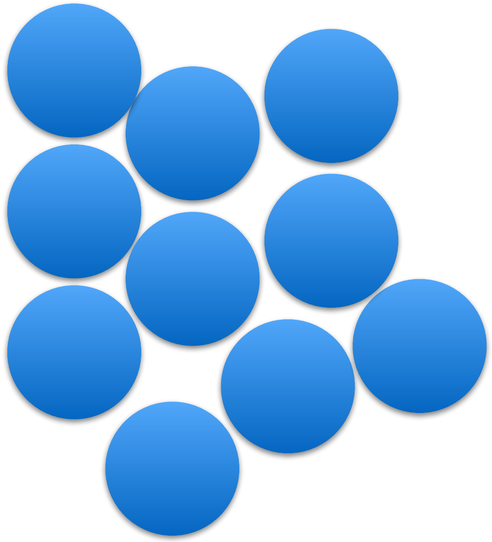


Molecular dynamics



$$\mathbf{F} = m\mathbf{a}$$

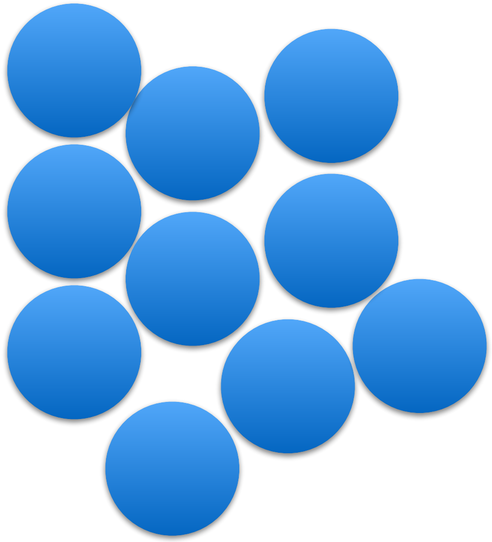
Molecular dynamics



$$\mathbf{F} = m\mathbf{a}$$

Specify interparticle forces: “force field”

Molecular dynamics



Numerically integrate particle positions

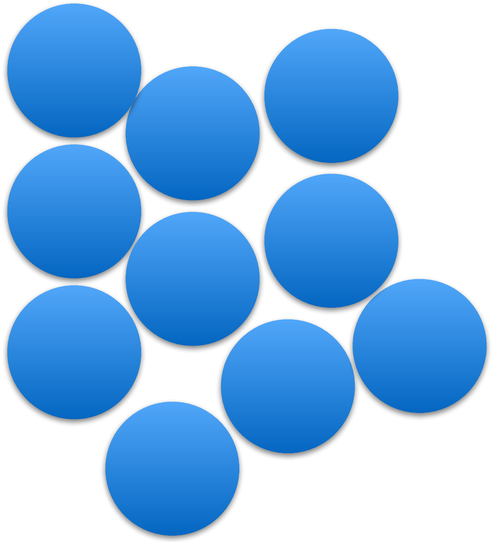
$$\mathbf{F} = m\mathbf{a}$$

Specify interparticle forces: “force field”

Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research

Molecular dynamics

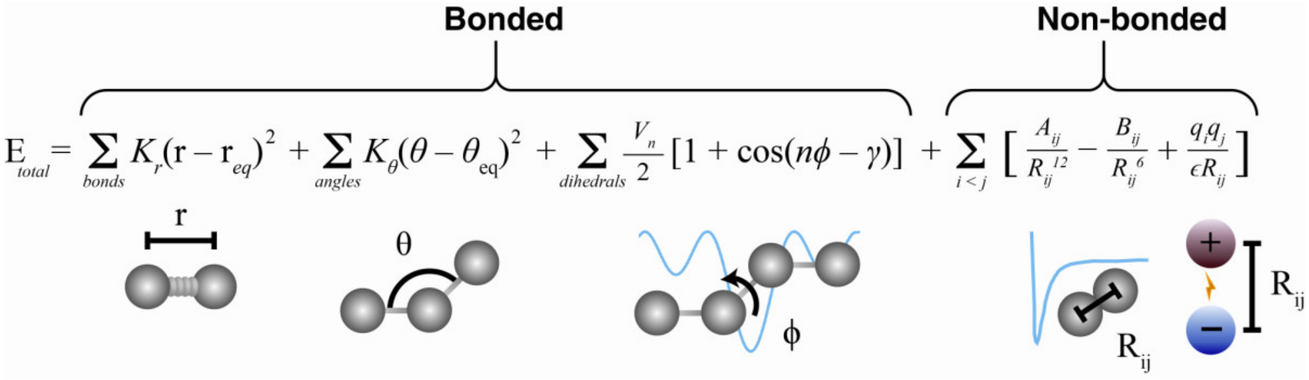
Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research



Numerically integrate particle positions

$$\mathbf{F} = m\mathbf{a}$$

Specify interparticle forces: “force field”

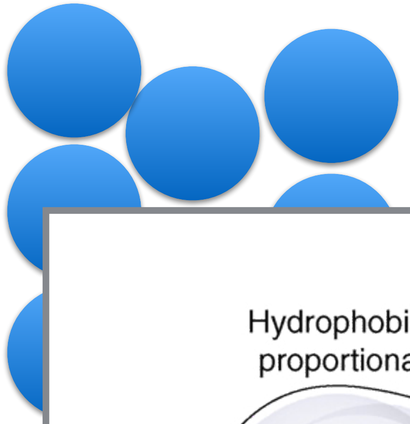


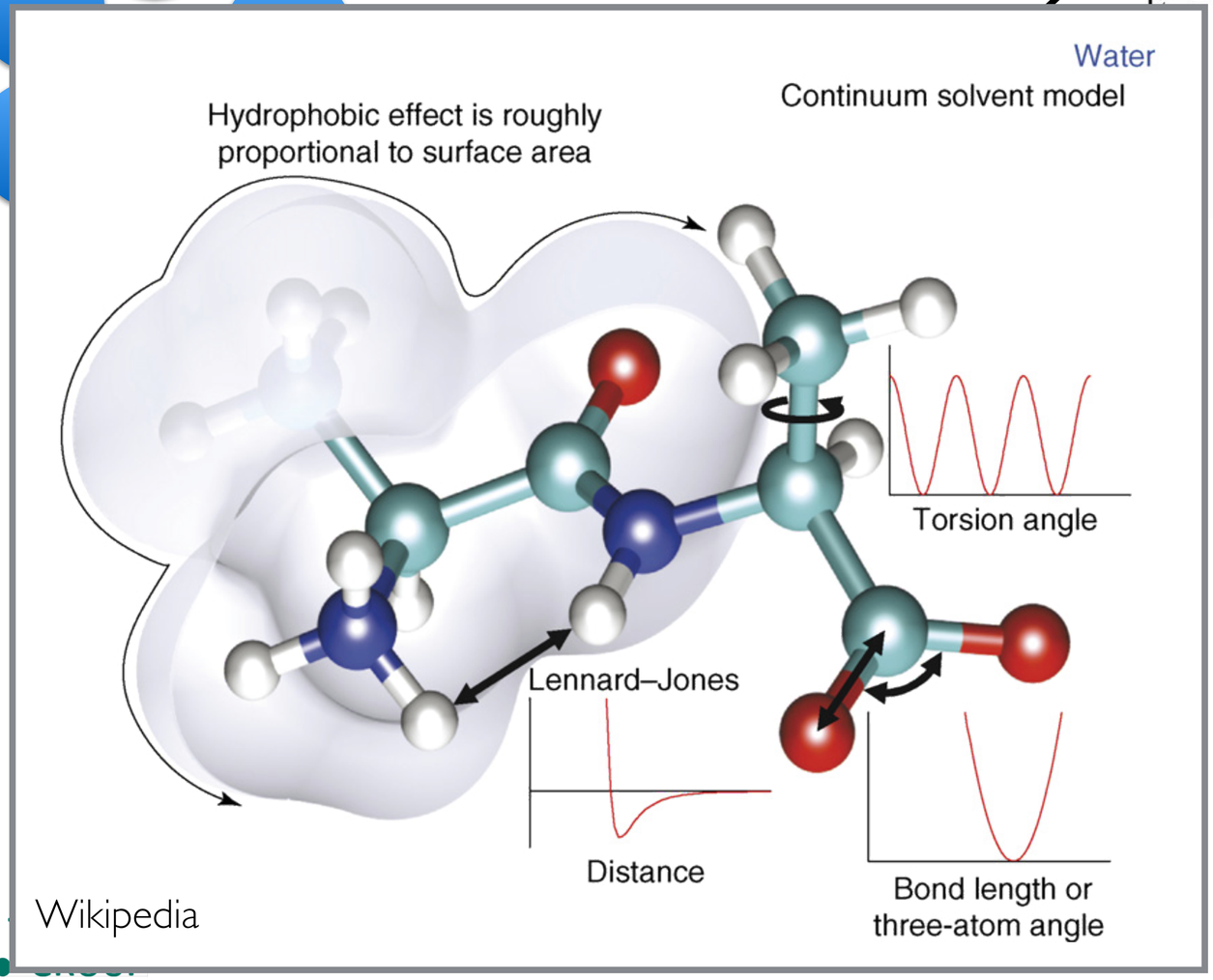
Durrant & McCammon, *BMC Biol* **9** (2011)

Molecular dynamics

Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research

Numerically integrate particle positions

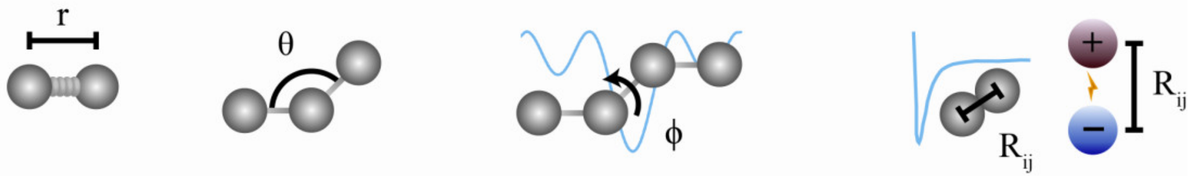




Wikipedia

Bonded **Non-bonded**

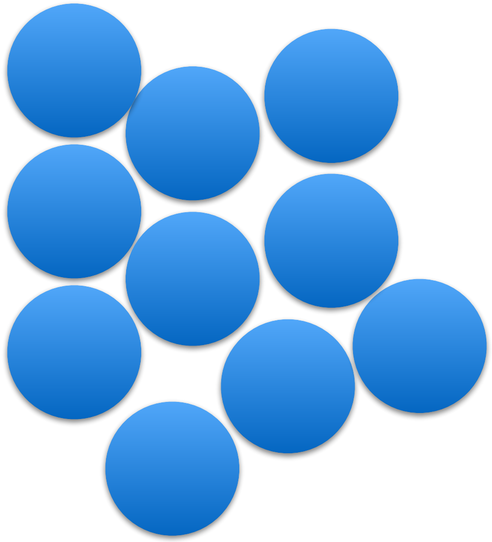
$$E = \sum_{bonds} K_r(r - r_{eq})^2 + \sum_{angles} K_\theta(\theta - \theta_{eq})^2 + \sum_{dihedrals} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right]$$



Durrant & McCammon, *BMC Biol* **9** (2011)

Molecular dynamics

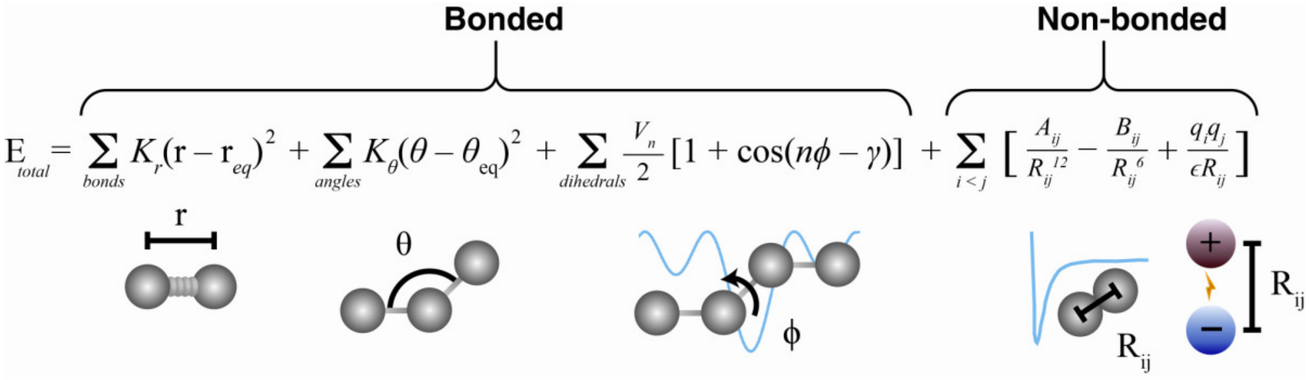
Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research



Numerically integrate particle positions

$$\mathbf{F} = m\mathbf{a}$$

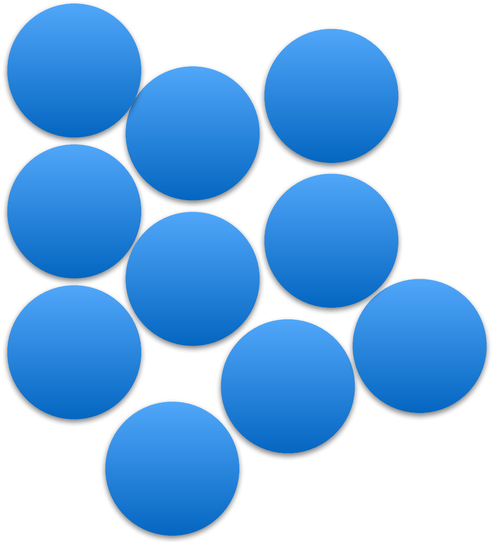
Specify interparticle forces: “force field”



Durrant & McCammon, *BMC Biol* **9** (2011)

Molecular dynamics

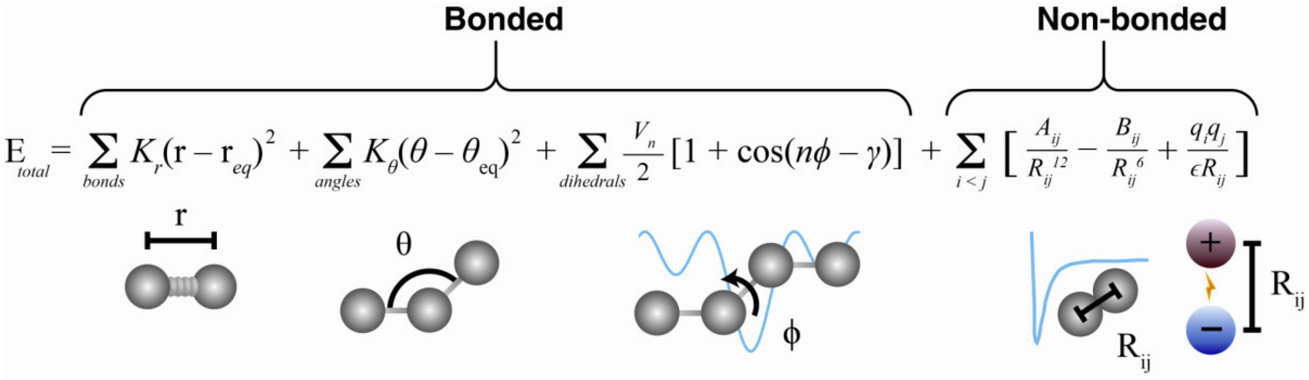
Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research



Numerically integrate particle positions

$$\mathbf{F} = m\mathbf{a}$$

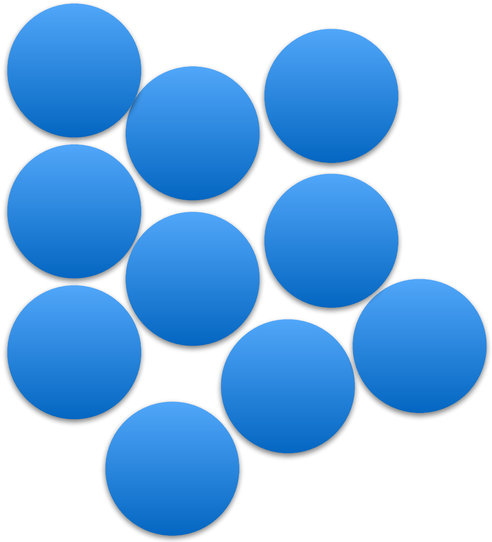
Specify interparticle forces: “force field”



Durrant & McCammon, *BMC Biol* **9** (2011)

Molecular dynamics

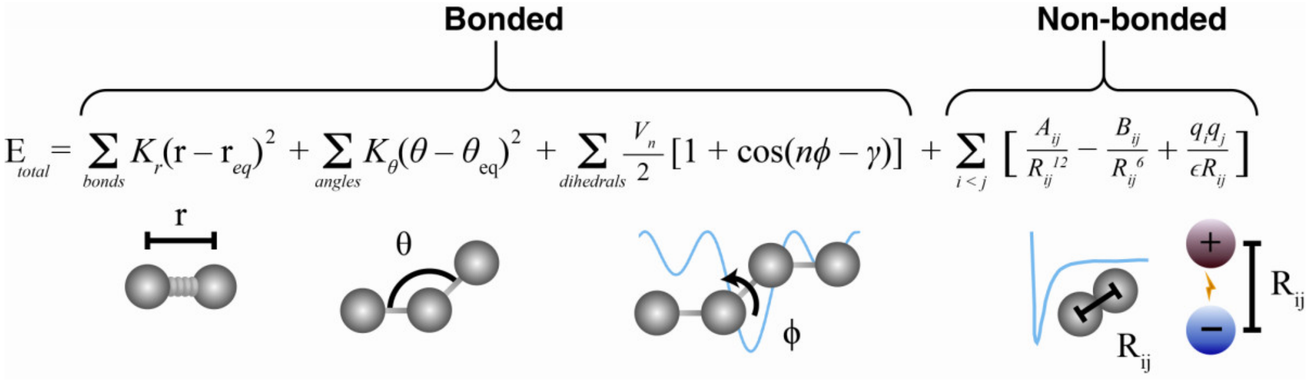
Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research



Numerically integrate particle positions

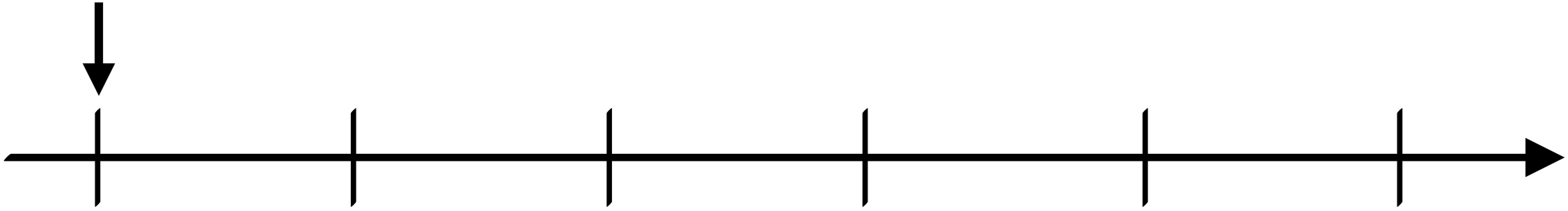
$$\mathbf{F} = m\mathbf{a}$$

Specify interparticle forces: “force field”



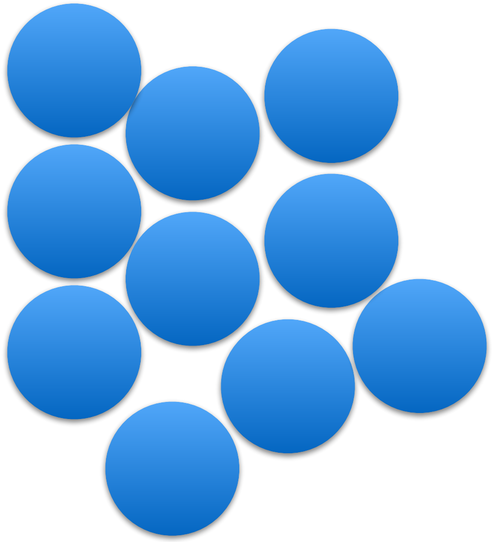
Durrant & McCammon, *BMC Biol* **9** (2011)

integration time step



Molecular dynamics

Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research



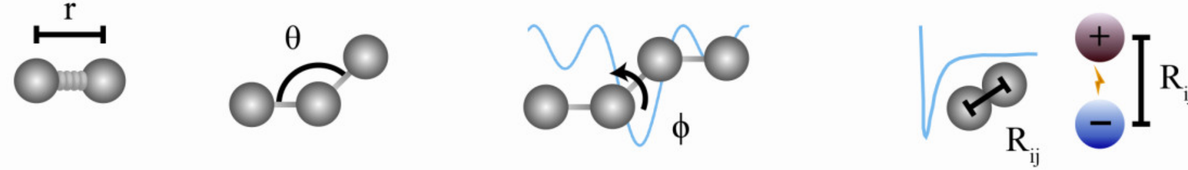
Numerically integrate particle positions

$$\mathbf{F} = m\mathbf{a}$$

Specify interparticle forces: “force field”

Bonded **Non-bonded**

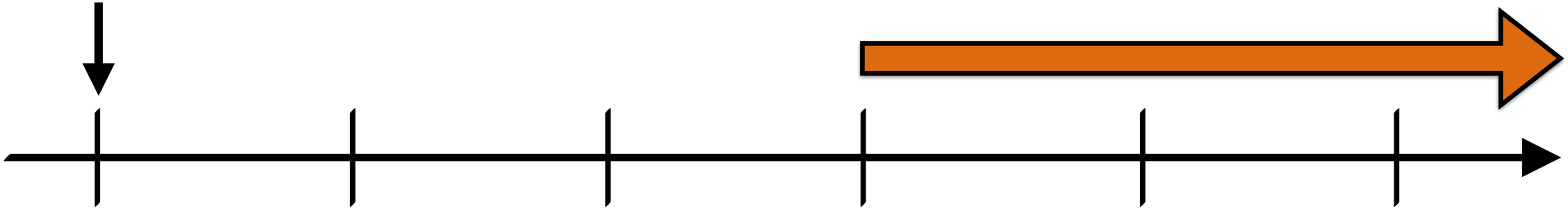
$$E_{total} = \sum_{bonds} K_r(r - r_{eq})^2 + \sum_{angles} K_\theta(\theta - \theta_{eq})^2 + \sum_{dihedrals} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right]$$



Durrant & McCammon, *BMC Biol* **9** (2011)

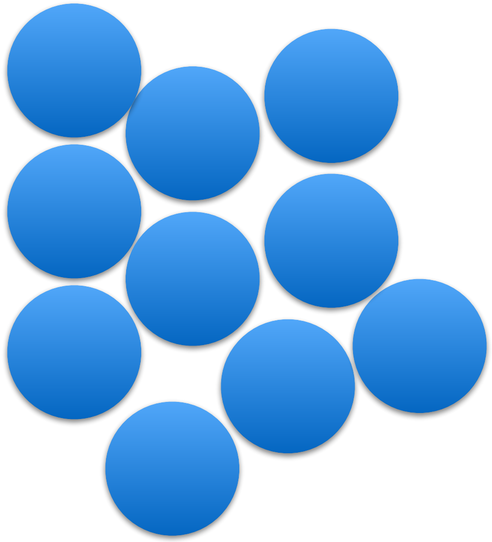
integration time step

Timescales of interest



Molecular dynamics

Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research



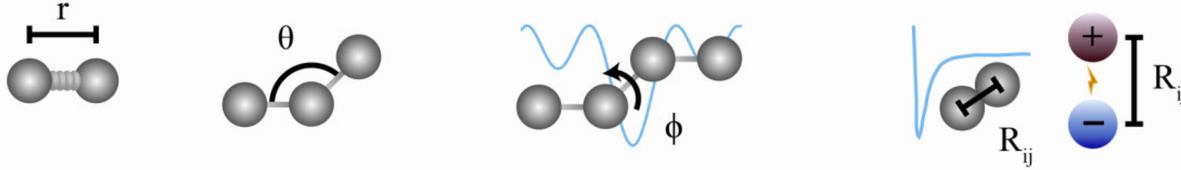
Numerically integrate particle positions

$$\mathbf{F} = m\mathbf{a}$$

Specify interparticle forces: “force field”

Bonded **Non-bonded**

$$E_{total} = \sum_{bonds} K_r(r - r_{eq})^2 + \sum_{angles} K_\theta(\theta - \theta_{eq})^2 + \sum_{dihedrals} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right]$$

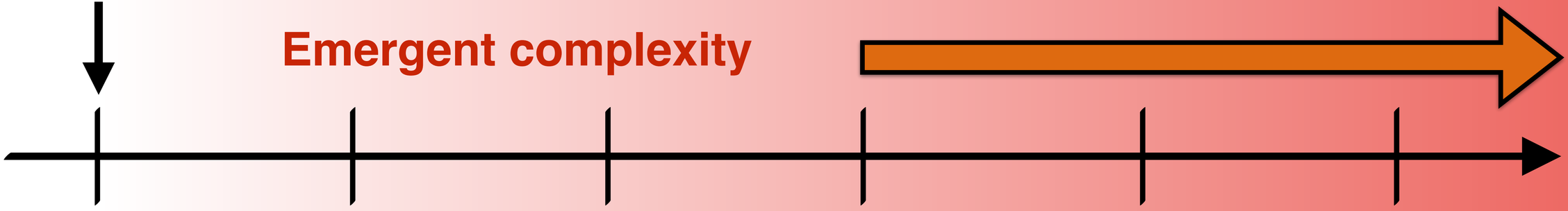


Durrant & McCammon, *BMC Biol* **9** (2011)

integration time step

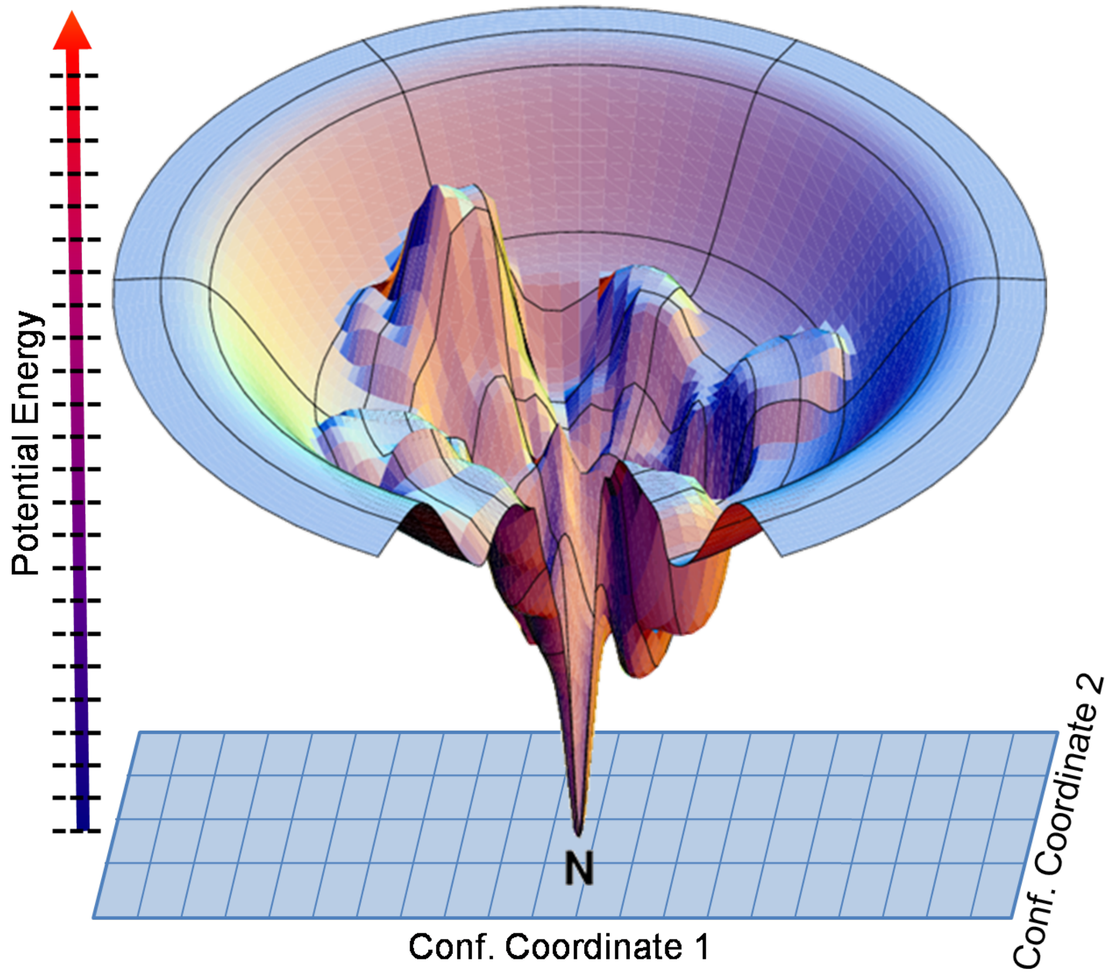
Timescales of interest

Emergent complexity



Links to machine learning

Potential energy surface

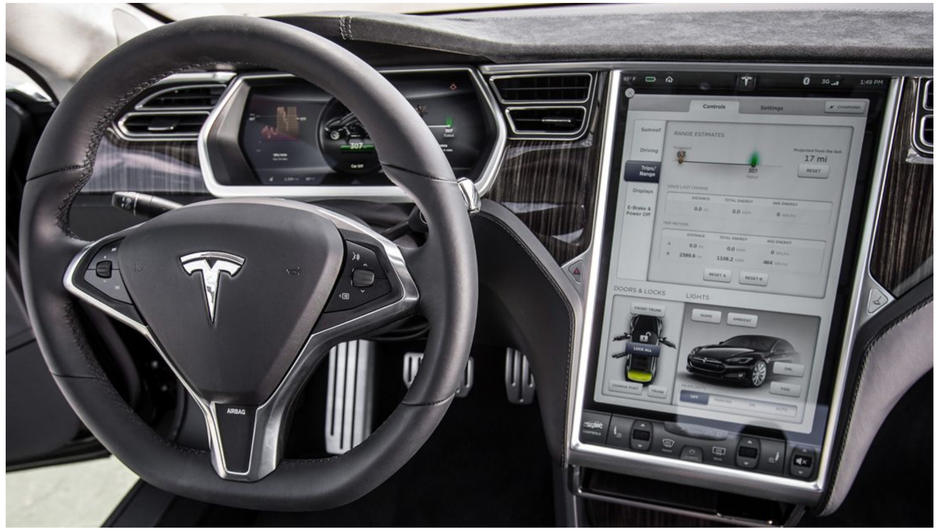


Can we build a more accurate PES?

Can we **easily** build an accurate PES?

Can we make the numerical integration faster and/or more efficient?

Disclaimer: only kernel methods covered



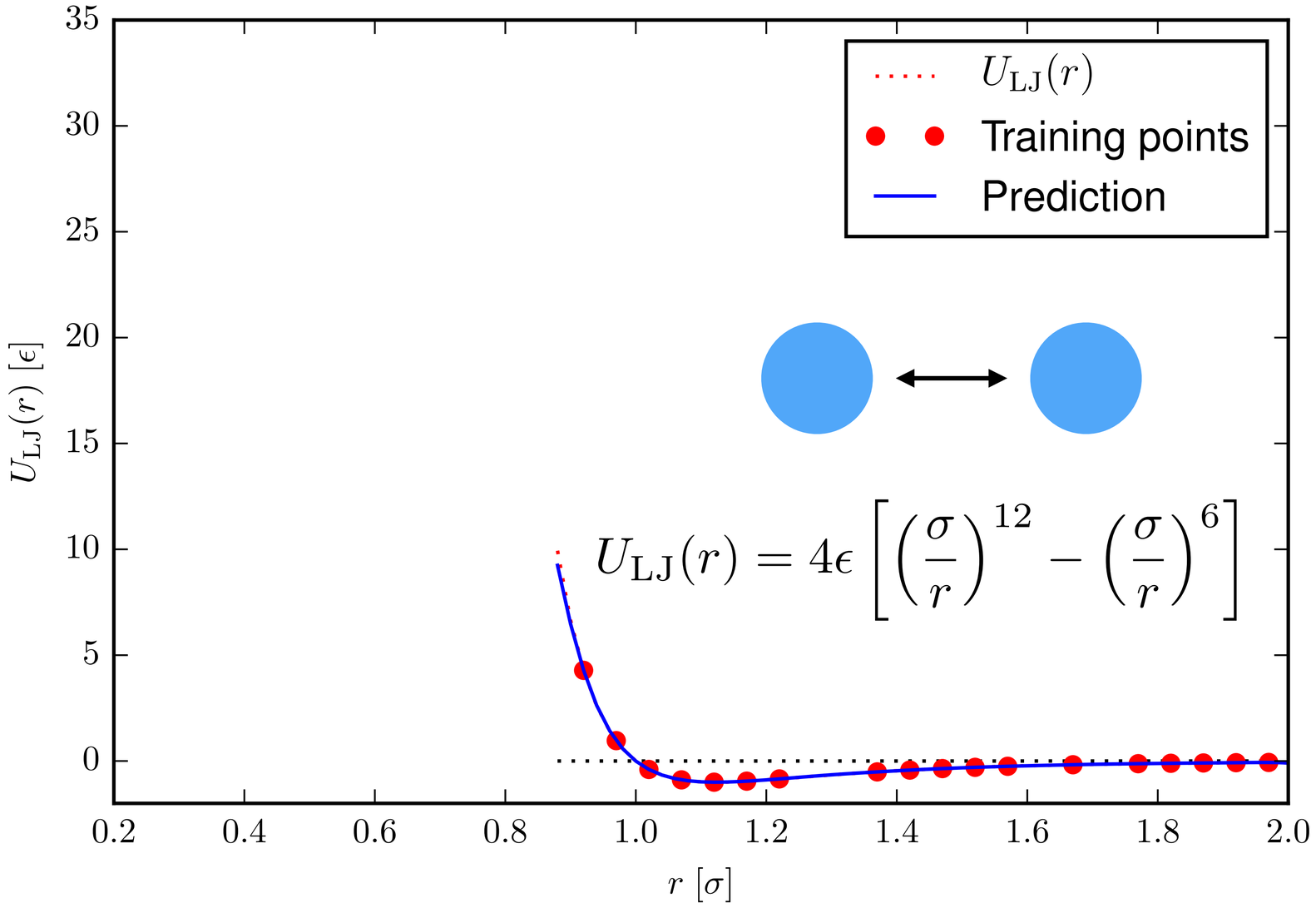
Kernel

- needs a representation
- linear algebra
- can be efficient with small data

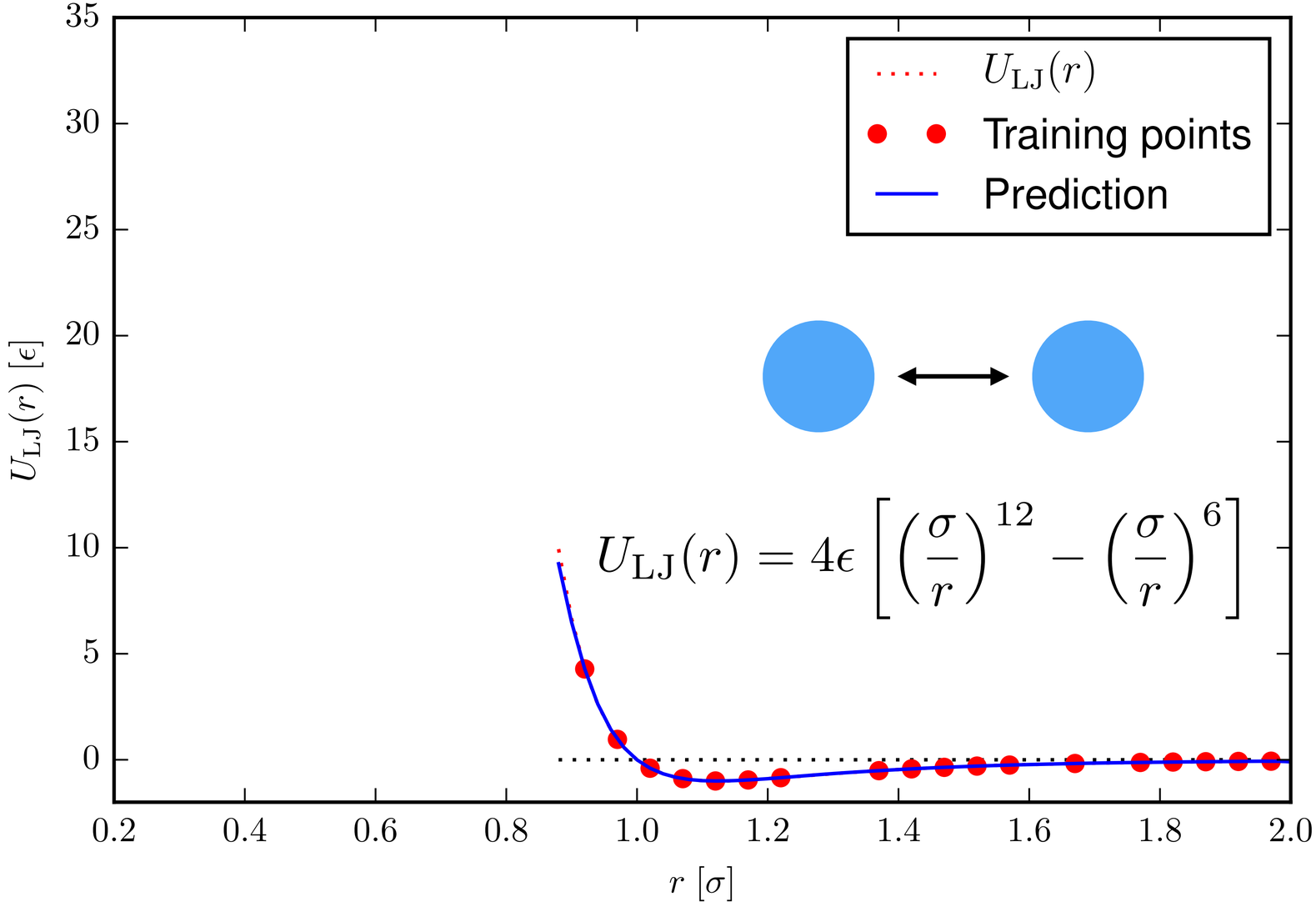
Deep learning

- learns the representation
- complex mathematical structure
- data hungry

Kernel machine learning 101



Kernel machine learning 101



Define kernel

$$K(r, r') = \exp\left(-\frac{(r - r')^2}{2\sigma^2}\right)$$

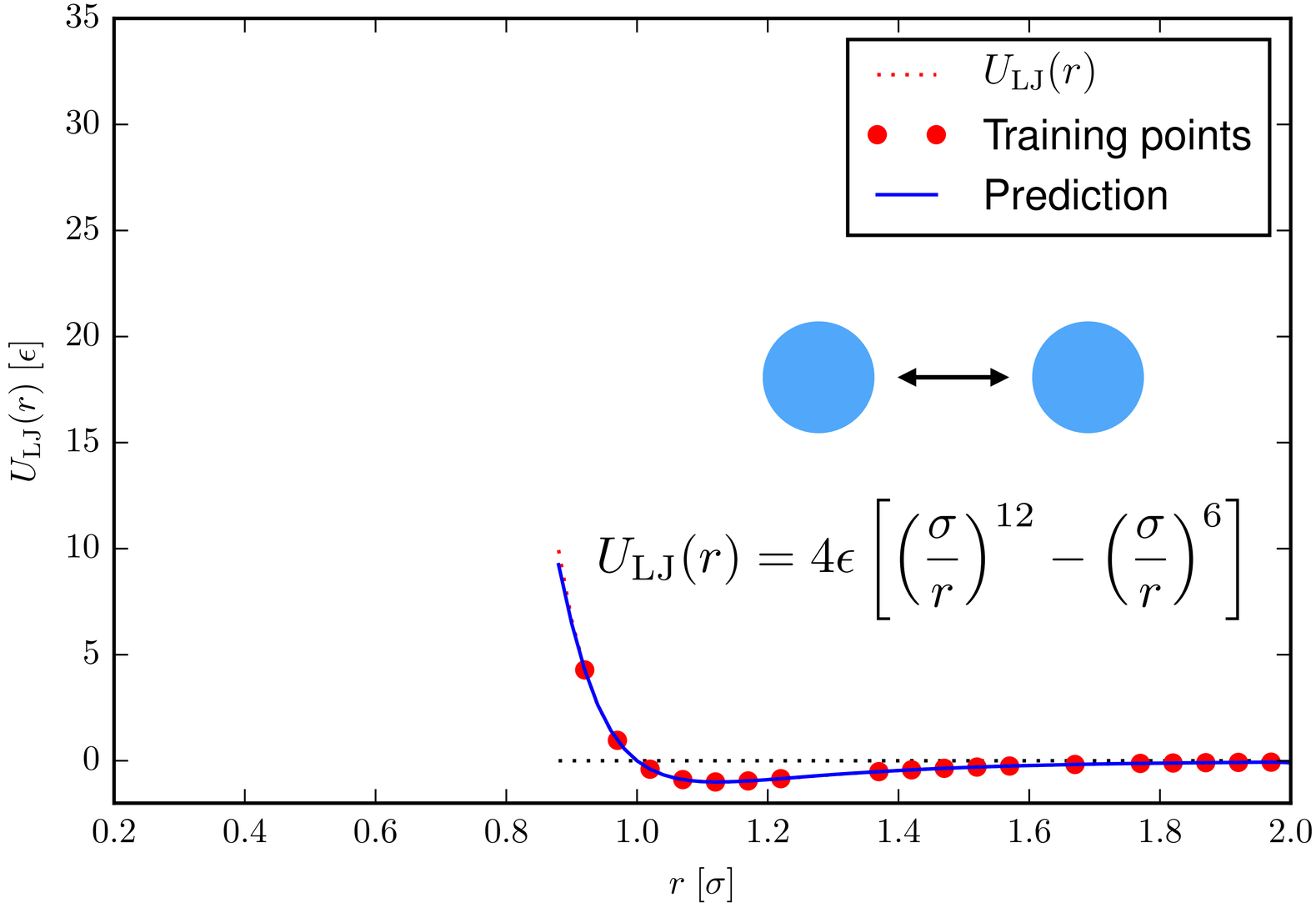
Make a prediction:

$$U(r) = \sum_i \alpha_i K(r_i^*, r)$$

Train your model:

$$\alpha = (K + \lambda \mathbb{I})^{-1} U$$

Kernel machine learning 101



Define kernel

$$K(r, r') = \exp\left(-\frac{(r - r')^2}{2\sigma^2}\right)$$

Make a prediction:

$$U(r) = \sum_i \alpha_i K(r_i^*, r)$$

Predicted energy

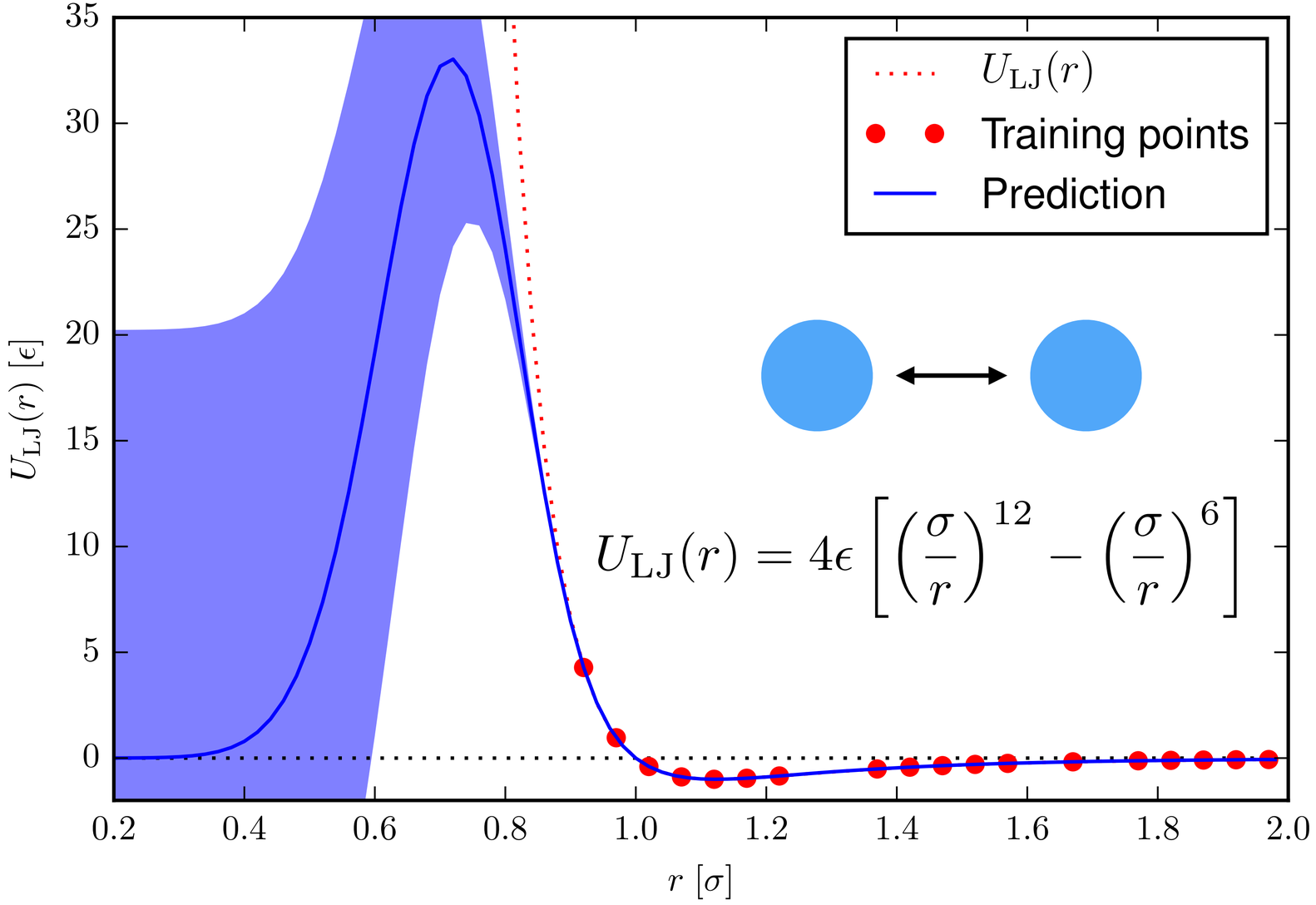
Training points

Query sample

Train your model:

$$\alpha = (K + \lambda \mathbb{I})^{-1} U$$

Extrapolation in machine learning



Define kernel

$$K(r, r') = \exp\left(-\frac{(r - r')^2}{2\sigma^2}\right)$$

Make a prediction:

$$U(r) = \sum_i \alpha_i K(r_i^*, r)$$

Predicted energy

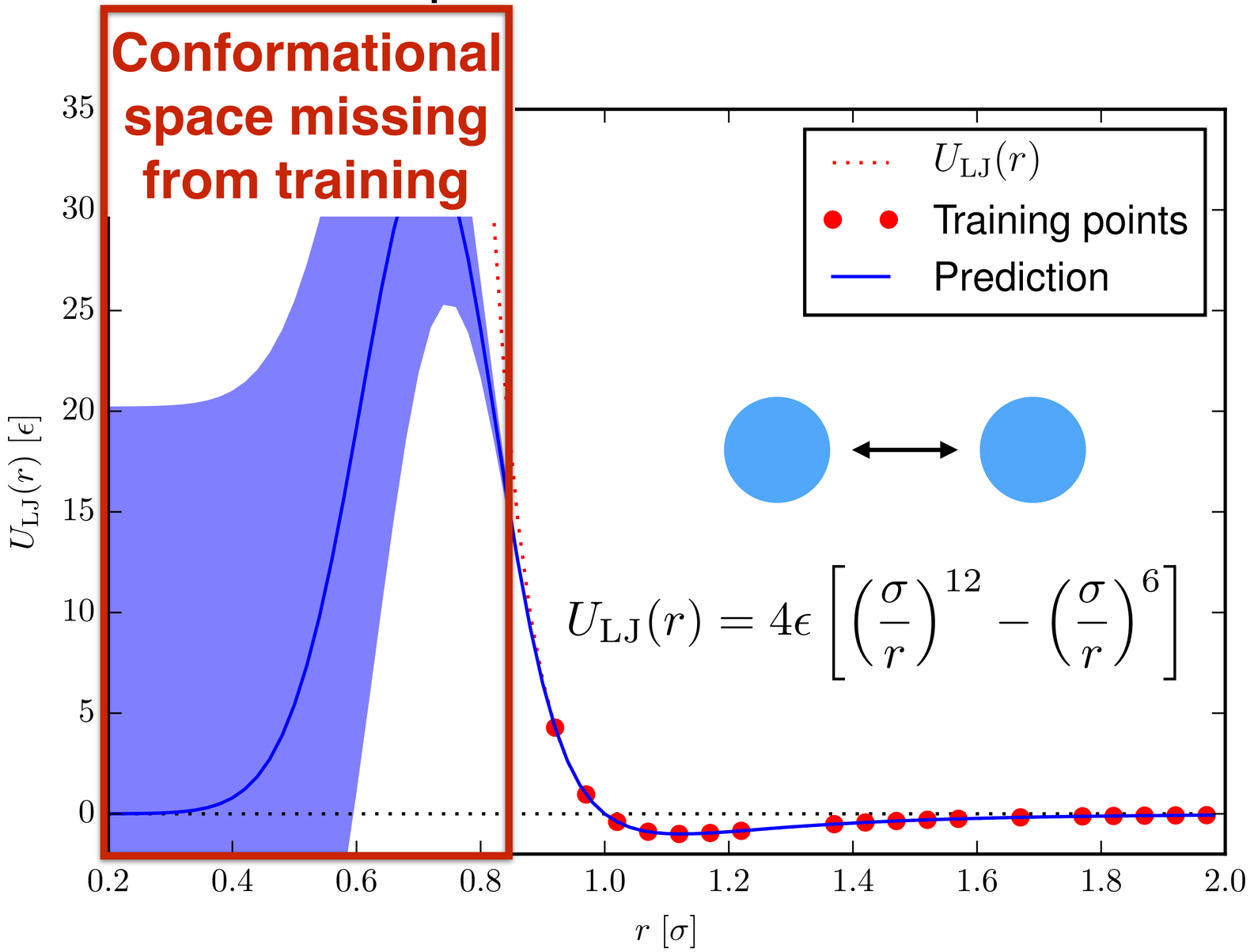
Training points

Query sample

Train your model:

$$\alpha = (K + \lambda \mathbb{I})^{-1} U$$

Extrapolation in machine learning



Define kernel

$$K(r, r') = \exp\left(-\frac{(r - r')^2}{2\sigma^2}\right)$$

Make a prediction:

$$U(r) = \sum_i \alpha_i K(r_i^*, r)$$

Predicted energy

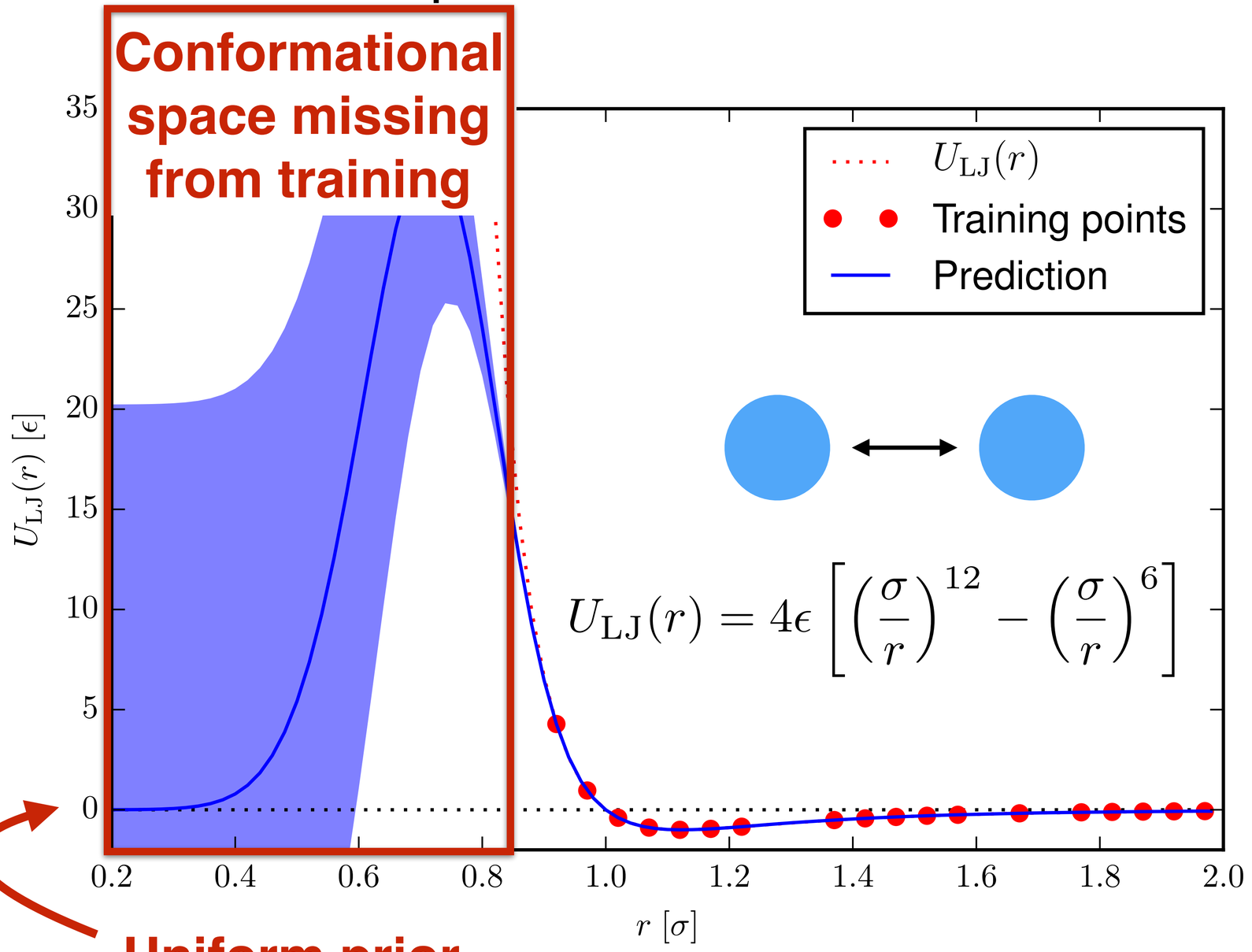
Training points

Query sample

Train your model:

$$\alpha = (K + \lambda \mathbb{I})^{-1} U$$

Extrapolation in machine learning



Define kernel

$$K(r, r') = \exp\left(-\frac{(r - r')^2}{2\sigma^2}\right)$$

Make a prediction:

$$U(r) = \sum_i \alpha_i K(r_i^*, r)$$

Predicted energy

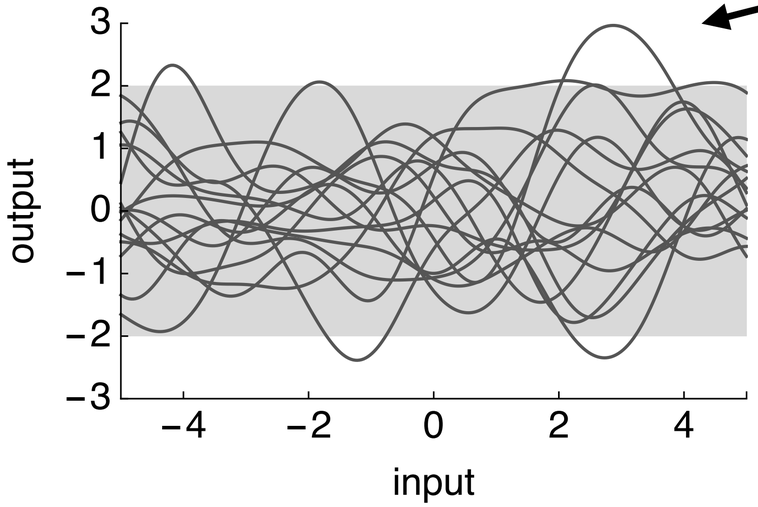
Training points

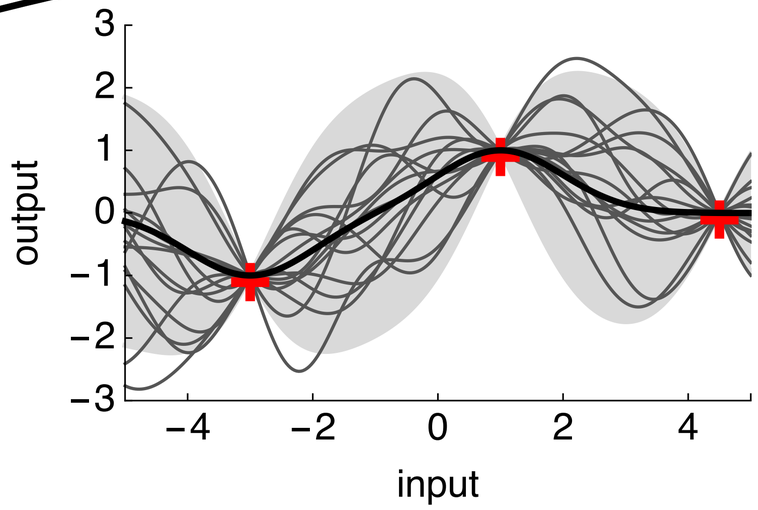
Query sample

Train your model:

$$\alpha = (K + \lambda \mathbb{I})^{-1} U$$

Bayesian inference





Bayes' formula

$$p(f,f^* \mid y) = \frac{p(y \mid f)\; p(f,f^*)}{p(y)}$$

likelihood

prior

posterior

normalization

Rasmussen, *Advanced lectures on machine learning*. Springer, 63-71 (2004)

Gaussian processes

$$f \sim \mathcal{GP}(m, k)$$

Rasmussen, *Advanced lectures on machine learning*. Springer, 63-71 (2004)

Gaussian processes

$$f \sim \mathcal{GP}(m, k)$$

$f(x)$

random variable: value of the stochastic function at *x*

$\mu_i = m(x_i)$

mean

$\Sigma_{ij} = k(x_i, x_j)$

covariance

Rasmussen, *Advanced lectures on machine learning*. Springer, 63-71 (2004)

Gaussian processes

$$f \sim \mathcal{GP}(m, k)$$

$f(x)$

random variable:
value of the
stochastic function
at *x*

$\mu_i = m(x_i)$

mean

$\Sigma_{ij} = k(x_i, x_j)$

covariance

kernel

$\mathbf{K}\alpha = p$

target property

Rasmussen, *Advanced lectures on machine learning*. Springer, 63-71 (2004)

Gaussian processes

$$f \sim \mathcal{GP}(m, k)$$

$f(x)$

random variable: value of the stochastic function at *x*

$\mu_i = m(x_i)$

mean

Nonuniform: replace by repulsive potential (Csanyi and coworkers, Clementi and Noé, …)

$\Sigma_{ij} = k(x_i, x_j)$

covariance

kernel

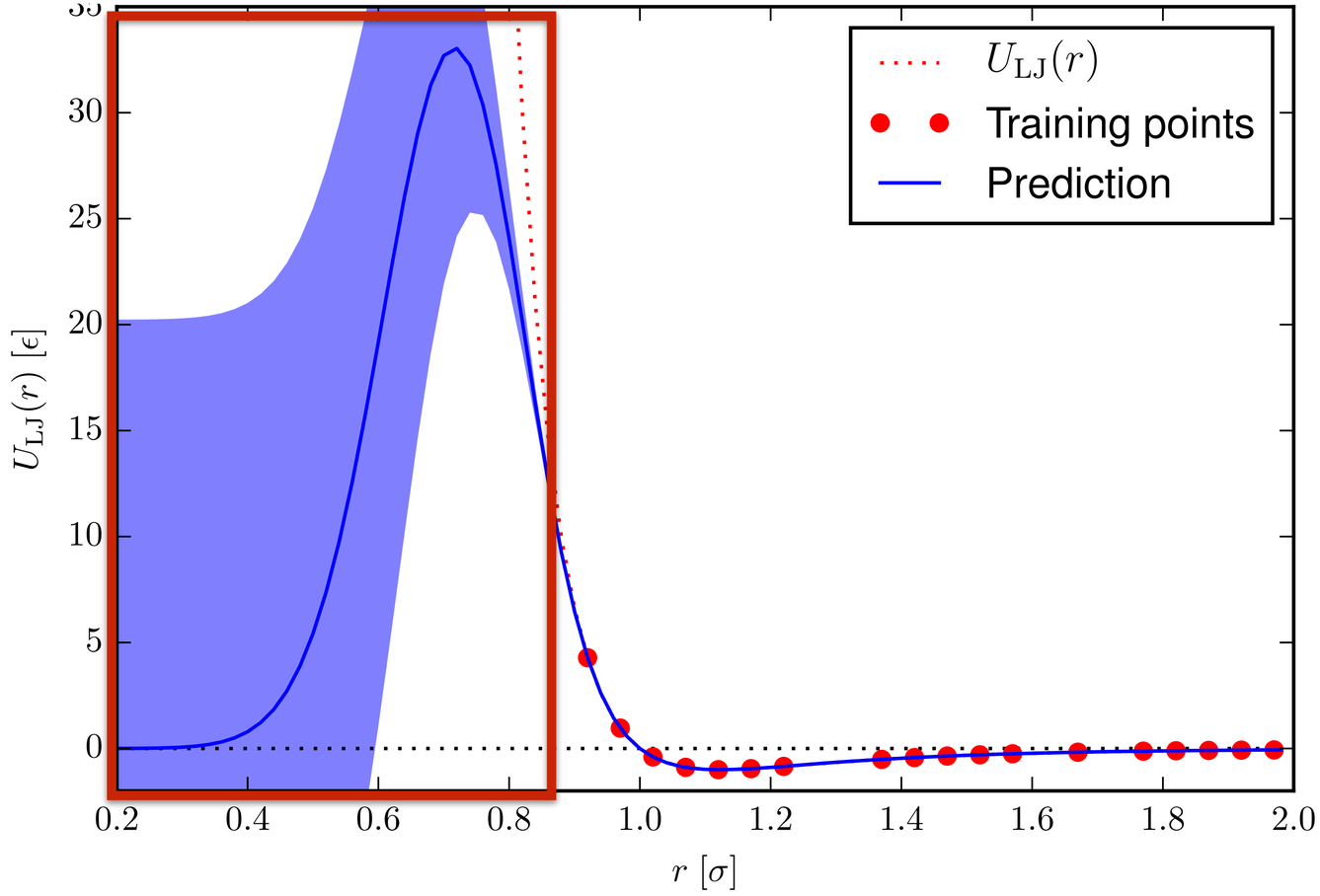
$\mathbf{K}\alpha = p$

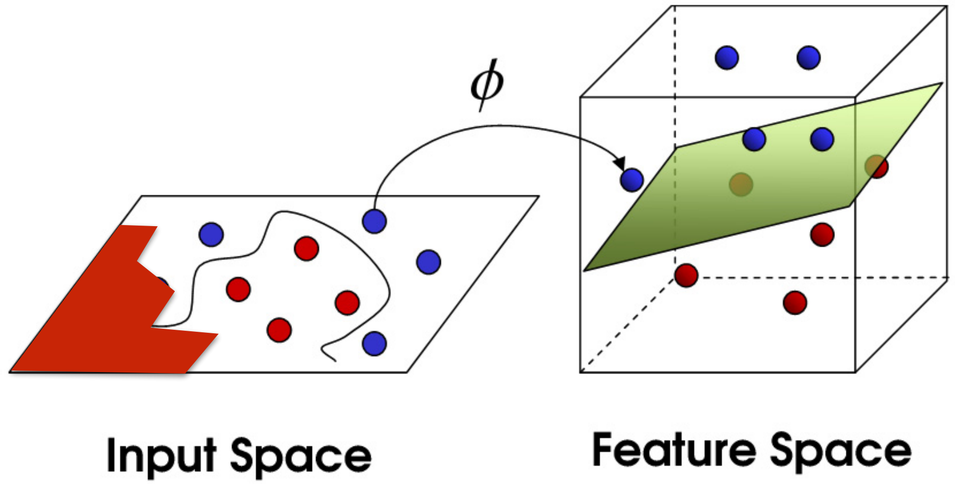
target property

Rasmussen, *Advanced lectures on machine learning*. Springer, 63-71 (2004)

Linking conformational and interpolation spaces

Conformational space missing from training





ML training set size is limited (kernels!)

Use physics to reduce the interpolation space

Symmetries and conservation laws

Mechanics 101: Principle of least action

$$\mathcal{S}[x(t)] = \int_{t_1}^{t_2} \mathrm{d}t\, L[x(t), \dot{x}(t), t]$$

action

microtrajectory

Lagrangian $L = T - V$

kinetic energy

potential energy

Mechanics 101: Principle of least action

$$\mathcal{S}[x(t)] = \int_{t_1}^{t_2} \mathrm{d}t\, L[x(t), \dot{x}(t), t]$$

action

microtrajectory

kinetic energy

potential energy

Lagrangian $L = T - V$

Hamilton's principle: system minimizes action (variational principle)

$$\dot{\mathcal{S}}[x^*(t)] = 0$$

Mechanics 101: Principle of least action

$$\mathcal{S}[x(t)] = \int_{t_1}^{t_2} \mathrm{d}t\, L[x(t), \dot{x}(t), t]$$

action

microtrajectory

kinetic energy

potential energy

Lagrangian $L = T - V$

Hamilton's principle: system minimizes action (variational principle)

$$\dot{\mathcal{S}}[x^*(t)] = 0$$

stationarity under small perturbations leads to Euler-Lagrange equations

$$\delta\mathcal{S} = \int_{t_1}^{t_2} \mathrm{d}t\, L(x^* + \varepsilon, \dot{x}^* + \dot{\varepsilon}, t) - L(x^*, \dot{x}^*, t)$$

$$= \int_{t_1}^{t_2} \mathrm{d}t \left(\varepsilon \frac{\partial L}{\partial x} + \dot{\varepsilon} \frac{\partial L}{\partial x} \right) = \int_{t_1}^{t_2} \mathrm{d}t \left(\varepsilon \frac{\partial L}{\partial x} - \varepsilon \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{x}} \right) = 0$$

Mechanics 101: Principle of least action

$$\mathcal{S}[x(t)] = \int_{t_1}^{t_2} \mathrm{d}t\, L[x(t), \dot{x}(t), t]$$

action

microtrajectory

kinetic energy

potential energy

Lagrangian $L = T - V$

Hamilton's principle: system minimizes action (variational principle)

$$\dot{\mathcal{S}}[x^*(t)] = 0$$

stationarity under small perturbations leads to Euler-Lagrange equations

$$\delta\mathcal{S} = \int_{t_1}^{t_2} \mathrm{d}t\, L(x^* + \varepsilon, \dot{x}^* + \dot{\varepsilon}, t) - L(x^*, \dot{x}^*, t)$$

$$= \int_{t_1}^{t_2} \mathrm{d}t \left(\varepsilon \frac{\partial L}{\partial x} + \dot{\varepsilon} \frac{\partial L}{\partial x} \right) = \int_{t_1}^{t_2} \mathrm{d}t \left(\varepsilon \frac{\partial L}{\partial x} - \varepsilon \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{x}} \right) = 0$$

integration by parts & $\varepsilon(t_1) = \varepsilon(t_2) = 0$

From symmetries, to invariants, to conserved quantities

$$\mathcal{S}[x(t), y(t), z(t)] = \int \mathrm{d}t \, \frac{m}{2} \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) - mgz$$

From symmetries, to invariants, to conserved quantities

$$\mathcal{S}[x(t), y(t), z(t)] = \int \mathrm{d}t \, \frac{m}{2} \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) - mgz$$

Introduce constant translations along x and y:

$$\begin{aligned} \mathcal{S}[x(t) + x_0, y(t) + y_0, z(t)] &= \int \mathrm{d}t \, \frac{m}{2} \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) - mgz \\ &= \mathcal{S}[x(t), y(t), z(t)] \end{aligned}$$

From symmetries, to invariants, to conserved quantities

$$\mathcal{S}[x(t), y(t), z(t)] = \int \mathrm{d}t \, \frac{m}{2} \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) - mgz$$

Introduce constant translations along x and y:

$$\mathcal{S}[x(t) + x_0, y(t) + y_0, z(t)] = \int \mathrm{d}t \, \frac{m}{2} \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) - mgz$$
$$= \mathcal{S}[x(t), y(t), z(t)]$$

(Translational) symmetry leaves the action invariant.
It leaves the Euler-Lagrange equation unchanged:

$$\frac{\partial L}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{x}} = 0$$

From symmetries, to invariants, to conserved quantities

$$\mathcal{S}[x(t), y(t), z(t)] = \int \mathrm{d}t \, \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) - mgz$$

Introduce constant translations along x and y:

$$\mathcal{S}[x(t) + x_0, y(t) + y_0, z(t)] = \int \mathrm{d}t \, \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) - mgz$$
$$= \mathcal{S}[x(t), y(t), z(t)]$$

(Translational) symmetry leaves the action invariant.
It leaves the Euler-Lagrange equation unchanged:

$$\frac{\partial L}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{x}} = 0$$

$$\frac{\partial L}{\partial x} = 0 \qquad \frac{\partial L}{\partial \dot{x}} = m\dot{x} = \mathrm{const}.$$

Translational invariance implies linear momentum conversation

From symmetries to conserved quantities (cont'd)

$$\mathcal{S}[\mathbf{r}(t)] = \int \mathrm{d}t \, \frac{m}{2}\dot{\mathbf{r}}^2 - V(r)$$

Rotational symmetry

Apply transformation $\mathbf{r} \to \mathbf{r}'$

where $\mathbf{r}'(t) = R\mathbf{r}(t) = \mathbf{r}(t) + \alpha \times \mathbf{r}(t)$

One can show that

$$\mathcal{S}[\mathbf{r}(t) + \alpha \times \mathbf{r}(t)] = \mathcal{S}[\mathbf{r}(t)]$$

Conservation of angular momentum

Time translation

Apply transformation $\mathbf{r} \to \mathbf{r}'$

where $\mathbf{r}'(t + \epsilon) = \mathbf{r}(t)$

One can show that

$$\mathcal{S}[\mathbf{r}'(t + \epsilon)] = \mathcal{S}[\mathbf{r}(t)]$$

(up to a boundary term)

Conservation of energy

Bañados & Reyes, *arXiv:1601.03616v3*

Noether's theorem

To every differentiable symmetry generated by local actions there corresponds a conserved quantity

3 examples:

- Translational symmetry: **Linear momentum conservation**
- Rotational symmetry: **Angular momentum conservation**
- Time translation: **Energy conservation**

THEORY GROUP

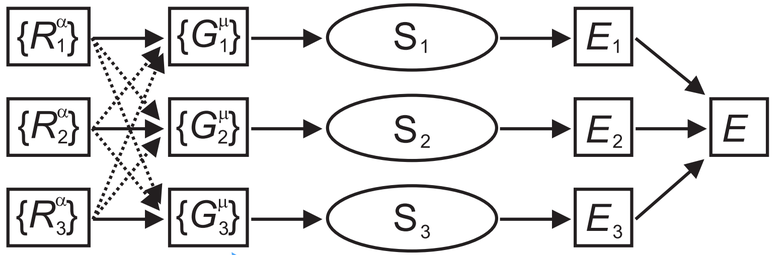
2 ways of encoding symmetries:

- Representation
- ML model

Encoding symmetries in the representation

Translational and rotational symmetries

Behler-Parrinello



Cartesian coordinates

Symmetry functions

$$G_i^1 = \sum_{j \neq i}^{\text{all}} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij})$$

Distances

$$G_i^2 = 2^{1-\zeta} \sum_{j,k \neq i}^{\text{all}} (1 + \lambda \cos\theta_{ijk})^\zeta \times e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk})$$

Angles

Behler & Parrinello, *Phys Rev Lett* **98** (2007)

Coulomb matrix

$$C_{ij} = \begin{cases} \frac{1}{2} Z_i^{2.4} & \forall \ i = j \\ \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} & \forall \ i \neq j \end{cases}$$

Distances

Rupp, Tkatchenko, Müller, von Lilienfeld, *Phys Rev Lett*, **108** (2012)

Encoding symmetries in the ML model

Encoding symmetries in ML models using group theory

Action of group G on input sample $x \mapsto T_g(x)$

Risi Kondor, *Group theoretical methods in machine learning, PhD thesis* (2008)

Encoding symmetries in ML models using group theory

Action of group G on input sample $x \mapsto T_g(x)$

Can we find a kernel that is invariant to this group action? $f(T_g(x)) = f(x) \forall g \in G$

$$k(x, x') = k(T_g(x), T_{g'}(x'))$$

Risi Kondor, *Group theoretical methods in machine learning, PhD thesis* (2008)

Encoding symmetries in ML models using group theory

Action of group G on input sample $x \mapsto T_g(x)$

Can we find a kernel that is invariant to this group action? $f(T_g(x)) = f(x) \forall g \in G$

$$k(x, x') = k(T_g(x), T_{g'}(x'))$$

To ensure invariance, symmetrize the kernel

$$k^G(x, x') = \frac{1}{|G|} \sum_{g \in G} k(x, T_g(x'))$$

Risi Kondor, *Group theoretical methods in machine learning, PhD thesis* (2008)

Example of symmetrized kernel

SOAP kernel/representation*

$$S(\hat{R}) \equiv S(\rho, \hat{R}\rho') = \int \mathrm{d}\mathbf{r}\, \rho(\mathbf{r})\rho'(\hat{R}\mathbf{r}) = \sum_{i,i'} \sum_{\substack{l,m \\ l',m',m''}} D^{l'}_{m'm''}(\hat{R}) \int \mathrm{d}r\, c^{i*}_{lm}(r) c^{i'}_{l'm'}(r) \int \mathrm{d}\hat{\mathbf{r}}\, Y^*_{lm}(\hat{\mathbf{r}}) Y_{l'm''}(\hat{\mathbf{r}})$$

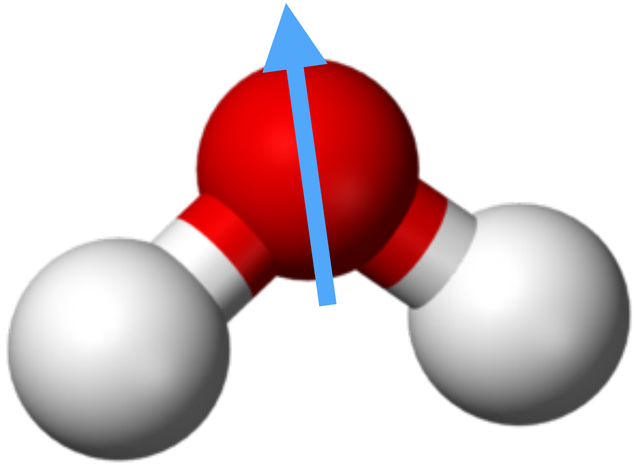
$$k(\rho, \rho') = \int \left| S(\rho, \hat{R}\rho') \right|^n \mathrm{d}\hat{R} =$$

$$= \int \mathrm{d}\hat{R} \left| \int \rho(\mathbf{r})\rho'(\hat{R}\mathbf{r})\mathrm{d}\mathbf{r} \right|^n$$

*Smooth Overlap of Atomic Positions: is a distance metric between two samples

Invariant vs. covariant properties

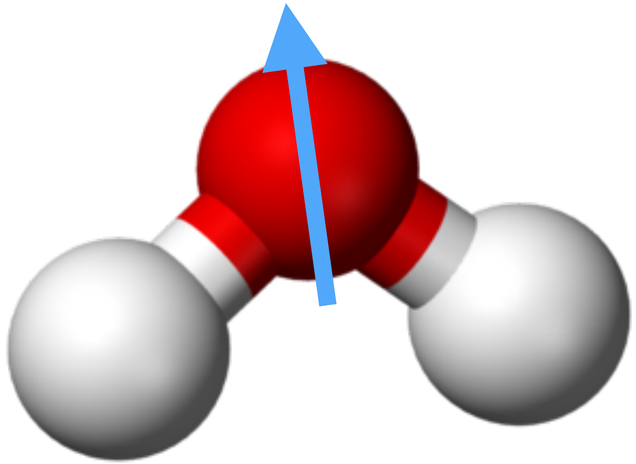
Tensorial property (e.g., dipole moment, force) **rotates** with the sample

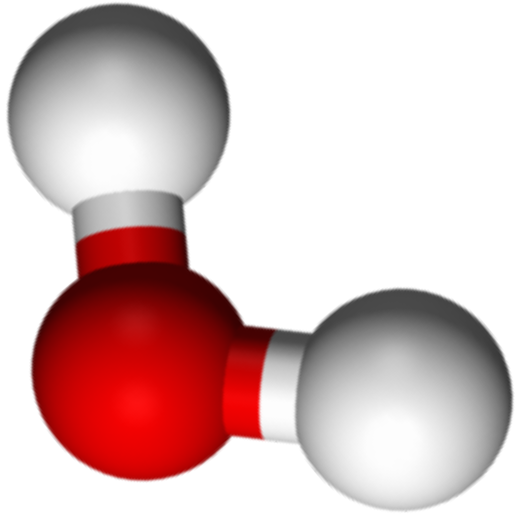


Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Invariant vs. covariant properties

Tensorial property (e.g., dipole moment, force) **rotates** with the sample

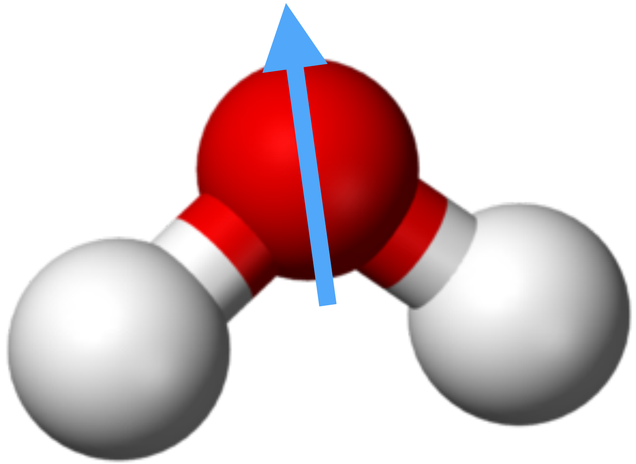


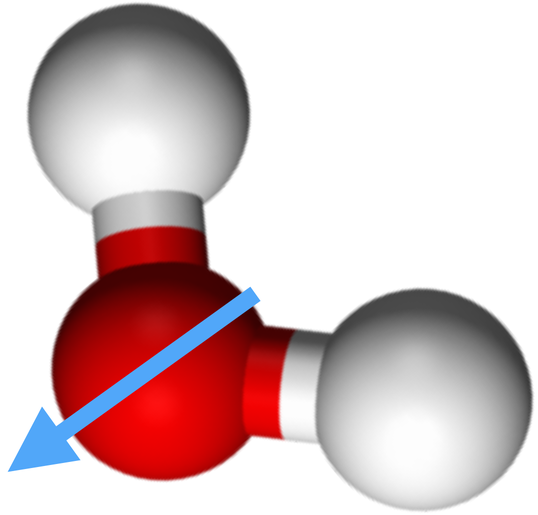


Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Invariant vs. covariant properties

Tensorial property (e.g., dipole moment, force) **rotates** with the sample

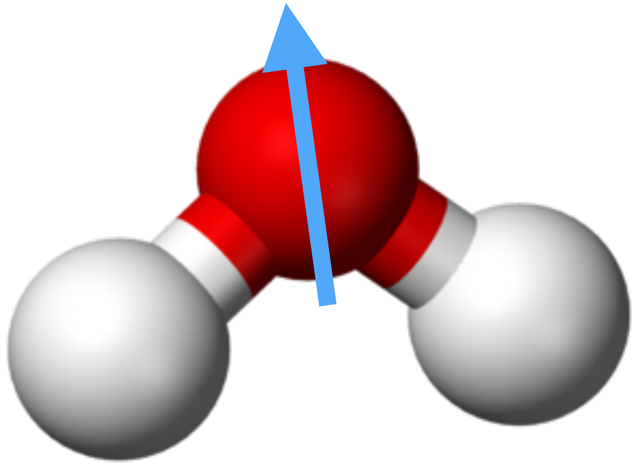


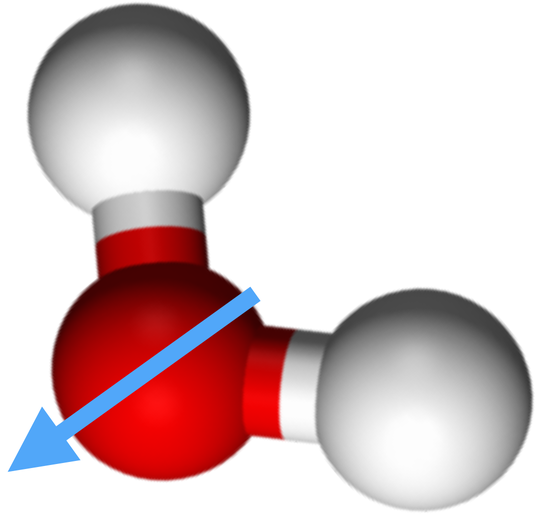


Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Invariant vs. covariant properties

Tensorial property (e.g., dipole moment, force) **rotates** with the sample





"Build kernel so as to encode the rotational properties of the target property"

Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Covariant kernels

Encode rotational properties of the target property in the **kernel**

Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Covariant kernels

Encode rotational properties of the target property in the **kernel**

$$\hat{\mathbf{f}}(\mathcal{S}\rho \mid \mathcal{D}) = \mathbf{S}\hat{\mathbf{f}}(\rho \mid \mathcal{D})$$

Force prediction

Descriptor

Training data

Transformation (rotation/inversion)

Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Covariant kernels

Encode rotational properties of the target property in the **kernel**

$$\hat{\mathbf{f}}(\mathcal{S}\rho \mid \mathcal{D}) = \mathbf{S}\hat{\mathbf{f}}(\rho \mid \mathcal{D})$$

Force prediction

Descriptor

Training data

Transformation (rotation/inversion)

"Transform the configuration, and the prediction transforms with it"

Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Covariant kernels

$$\mathbf{K}(\mathcal{S}\rho, \mathcal{S}'\rho') = \mathbf{S}\mathbf{K}(\rho, \rho')\mathbf{S}'^{\mathrm{T}}$$

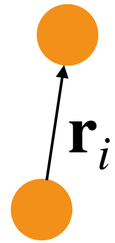
Kernel

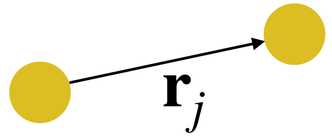
Configurations

Transformations (rotation/inversion)

$$\mathbf{K}(\rho, \rho') = \int d\mathcal{R}\, \mathbf{R} k_b(\rho, \mathcal{R}\rho')$$

$$\mathbf{K}^{\mu}(\rho, \rho') = \frac{1}{L}\sum_{ij} \phi(r_i, r_j)\mathbf{r}_i \otimes \mathbf{r}_j'^{\mathrm{T}}$$





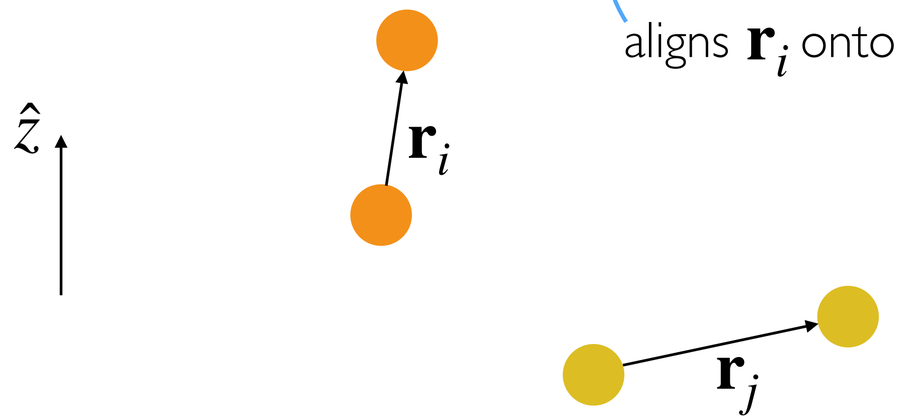
Glielmo, Sollich, De Vita, *Phys Rev B* **95** (2017)

Extension to higher-order tensors

$$\mathbf{K}^{\mathrm{Q}}(\mathcal{S}\rho, \mathcal{S}'\rho') = \mathbf{S}'\mathbf{S}^{\mathrm{T}}\mathbf{K}^{\mathrm{Q}}(\rho, \rho')\mathbf{S}\mathbf{S}'^{\mathrm{T}}$$

$$\mathbf{K}^{\mathrm{Q}}(\rho, \rho') = \frac{1}{L}\sum_{ij}\left(\mathbf{R}_j^{\mathrm{T}} \otimes \mathbf{R}_i\right)\Phi(r_i, r_j)\left(\mathbf{R}_i{}^{\mathrm{T}} \otimes \mathbf{R}_j\right)$$

aligns $\mathbf{r}_i$ onto $\hat{z}$



$$\Phi(r_i, r_j) = \mathrm{e}^{\frac{-\alpha_{ij}^2}{4\sigma^2}}\int \mathrm{d}\alpha \int \mathrm{d}\beta \int \mathrm{d}\gamma \frac{\sin\beta}{8\pi^2} \times \mathbf{R}^{\mathrm{T}}(\alpha, \beta, \gamma) \otimes \mathbf{R}(\alpha, \beta, \gamma)\mathrm{e}^{\frac{r_i r_j \cos\beta}{2\sigma^2}}$$

$$= \begin{pmatrix} \varphi_1 & 0 & 0 & 0 & \varphi_2 & 0 & 0 & 0 & 0 \\ 0 & \varphi_1 & 0 & -\varphi_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \varphi_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\varphi_2 & 0 & \varphi_1 & 0 & 0 & 0 & 0 & 0 \\ \varphi_2 & 0 & 0 & 0 & \varphi_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \varphi_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \varphi_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi_4 \end{pmatrix}$$

Bereau, DiStasio Jr., Tkatchenko, von Lilienfeld, *J Chem Phys* **148** (2018)
Grisafi, Wilkins, Csanyi, Ceriotti, Phys Rev Lett **120** (2018)

Application to intermolecular interactions

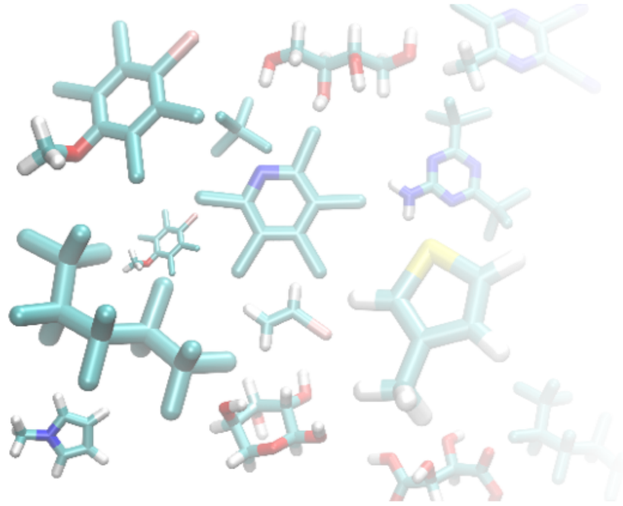
Transferable model for intermolecular interactions

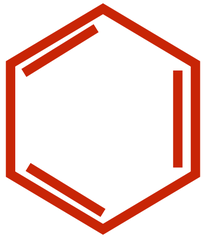
Physics-based models

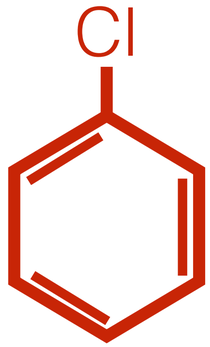
- Encode laws, symmetries
- Little chemical information

Data-driven models

- Need to *learn* laws, symmetries
- Interpolate across chemistry







Compositional (e.g., benzene vs. chlorobenzene)

Transferable model for intermolecular interactions

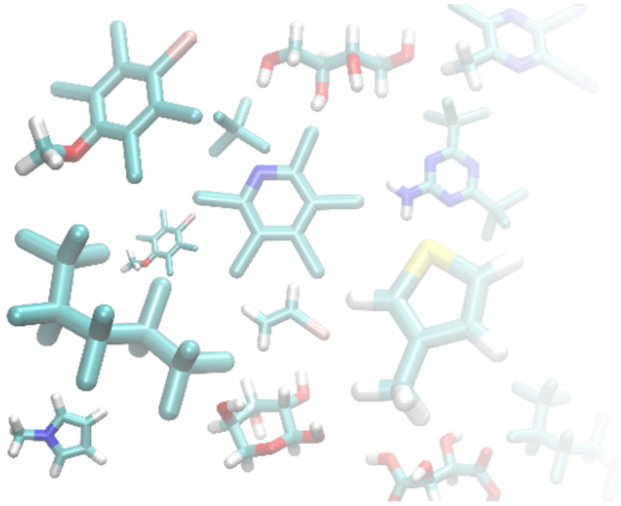
Physics-based models

- Encode laws, symmetries
- Little chemical information

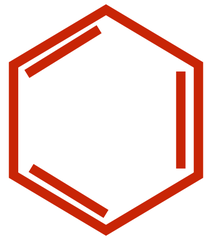
Data-driven models

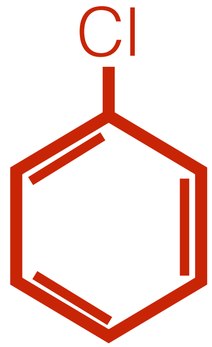
- Need to *learn* laws, symmetries
- Interpolate across chemistry

Any small molecule made of H, C, O, N neutral compounds



Compositional (e.g., benzene vs. chlorobenzene)





Physics-based aspect

Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research

Long-ranged

- Static electrostatics
- Many-body dispersion
- Polarization

Short-ranged

- Charge penetration
- Repulsion
- (Charge transfer)

Van Vleet, Misquitta, Stone, and Schmidt, *J. Chem. Theory Comput.* **12** (2016); Vandenbrande, et al., *J. Chem. Theory Comput.* **13** (2017); Grimme, *J. Chem. Theory Comput.* **10** (2014); Verstraelen, et al., *J. Chem. Theory Comput.* **12** (2016); Metz et al, *J. Chem. Theory Comput* **12** (2016)

Physics-based aspect

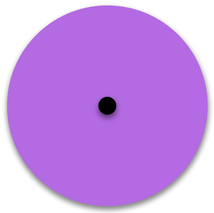
Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research

Long-ranged

Perturbation theory

- Static electrostatics
- Many-body dispersion
- Polarization

$$V_{\mathrm{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r}$$

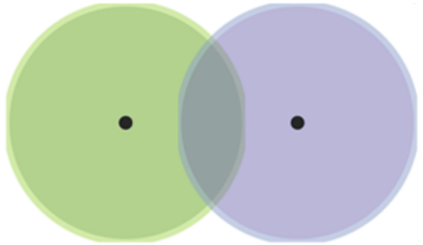


Short-ranged

Overlap models

- Charge penetration
- Repulsion
- (Charge transfer)

$$S_{ij} = \int \mathrm{d}^3\mathbf{r}\rho_i(\mathbf{r})\rho_j(\mathbf{r})$$



Van Vleet, Misquitta, Stone, and Schmidt, *J. Chem. Theory Comput.* **12** (2016); Vandenbrande, et al., *J. Chem. Theory Comput.* **13** (2017); Grimme, *J. Chem. Theory Comput.* **10** (2014); Verstraelen, et al., *J. Chem. Theory Comput.* **12** (2016); Metz et al, *J. Chem. Theory Comput* **12** (2016)

THEORY GROUP

Physics-based aspect

Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research

Long-ranged

- Static electrostatics
- Many-body dispersion
- Polarization

Short-ranged

- Charge penetration
- Repulsion
- (Charge transfer)

Use ML to predict atoms-in-molecules properties

- Multipole moments
- Hirshfeld ratios
- Atomic density widths/populations

Van Vleet, Misquitta, Stone, and Schmidt, *J. Chem. Theory Comput.* **12** (2016); Vandenbrande, et al., *J. Chem. Theory Comput.* **13** (2017); Grimme, *J. Chem. Theory Comput.* **10** (2014); Verstraelen, et al., *J. Chem. Theory Comput.* **12** (2016); Metz et al, *J. Chem. Theory Comput* **12** (2016)

Physics-based aspect

Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research

Long-ranged

- Static electrostatics
- Many-body dispersion
- Polarization

Multipole moments

Hirshfeld ratios

Short-ranged

- Charge penetration
- Repulsion
- (Charge transfer)

Atomic density widths/populations

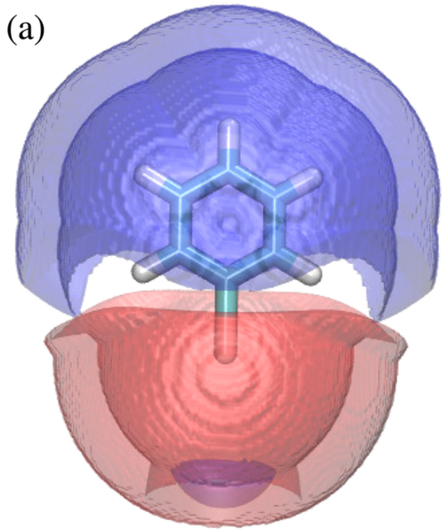
Use ML to predict atoms-in-molecules properties

- Multipole moments
- Hirshfeld ratios
- Atomic density widths/populations

Van Vleet, Misquitta, Stone, and Schmidt, *J. Chem. Theory Comput.* **12** (2016); Vandenbrande, et al., *J. Chem. Theory Comput.* **13** (2017); Grimme, *J. Chem. Theory Comput.* **10** (2014); Verstraelen, et al., *J. Chem. Theory Comput.* **12** (2016); Metz et al, *J. Chem. Theory Comput* **12** (2016)

Static multipole electrostatics

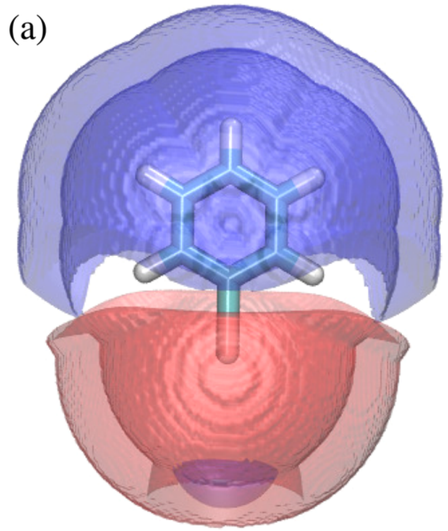
$$V_{\mathrm{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r}$$

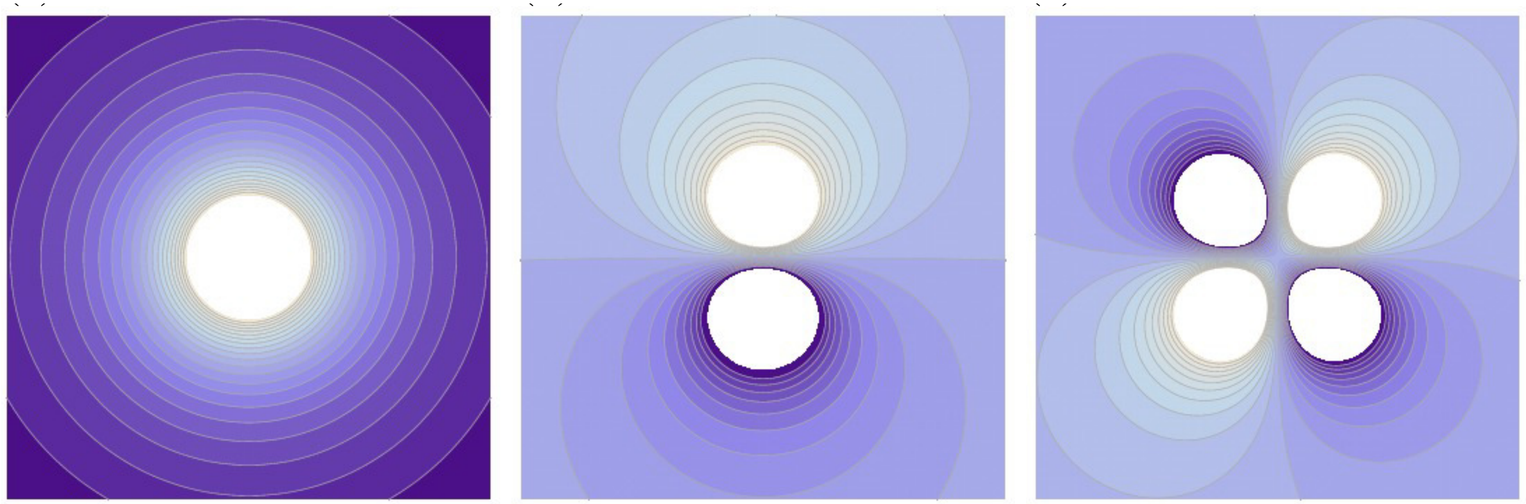


Stone, *The Theory of Intermolecular Forces*
Bereau and Meuwly, *Many-Body Effects and Electrostatics in Biomolecules*

Static multipole electrostatics

$$V_{\mathrm{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r}$$





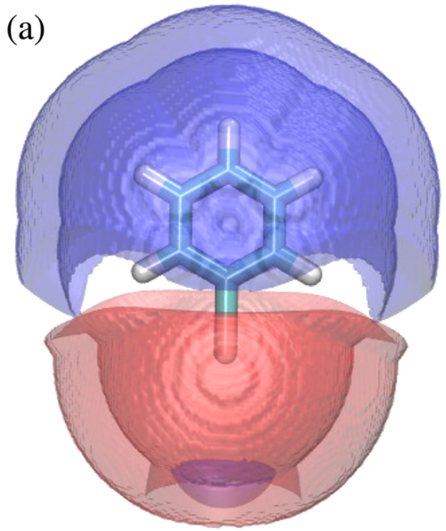
$$4\pi\varepsilon_0\Phi(\mathbf{r}) = \frac{q}{R} + \frac{\mu_\alpha R_\alpha}{R^3} + \frac{1}{3}\Theta_{\alpha\beta}\frac{3R_\alpha R_\beta - R^2\delta_{\alpha\beta}}{R^5} + \ldots$$

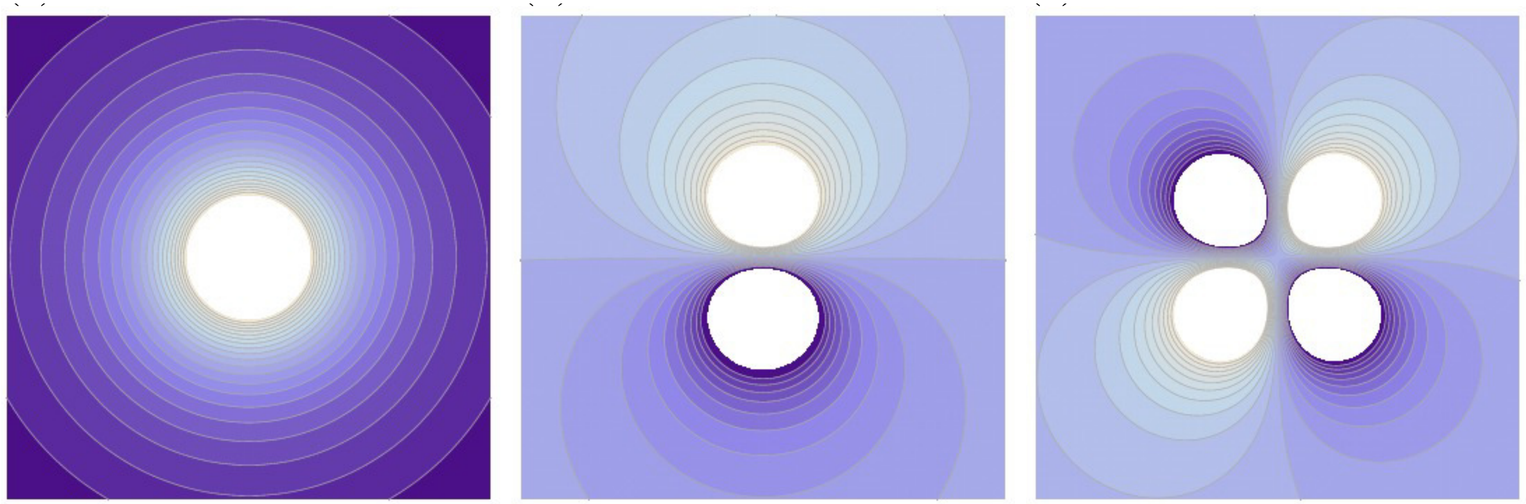
Stone, *The Theory of Intermolecular Forces*

Bereau and Meuwly, *Many-Body Effects and Electrostatics in Biomolecules*

Static multipole electrostatics

$$V_{\mathrm{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r}$$





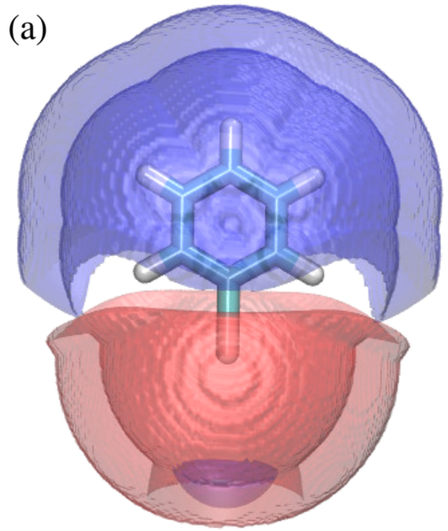
$$4\pi\varepsilon_0\Phi(\mathbf{r}) = \frac{q}{R} + \frac{\mu_\alpha R_\alpha}{R^3} + \frac{1}{3}\frac{\Theta_{\alpha\beta}\,3R_\alpha R_\beta - R^2\delta_{\alpha\beta}}{R^5} + \ldots$$

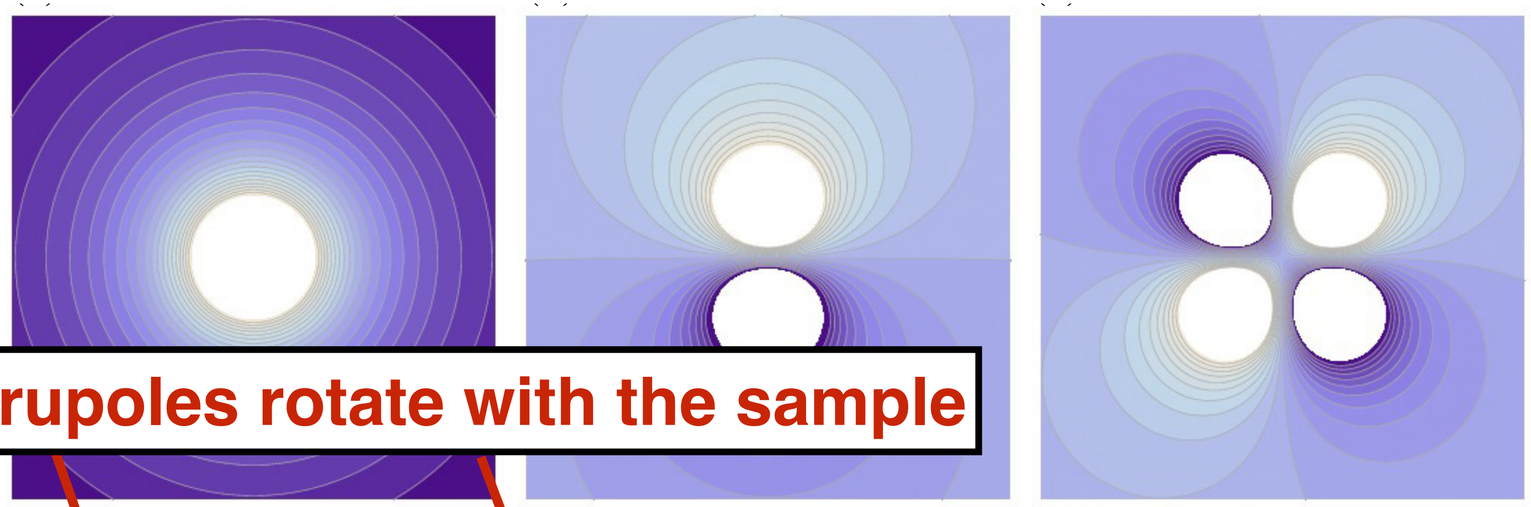
Stone, *The Theory of Intermolecular Forces*

Bereau and Meuwly, *Many-Body Effects and Electrostatics in Biomolecules*

Static multipole electrostatics

$$V_{\mathrm{Coulomb}}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r}$$





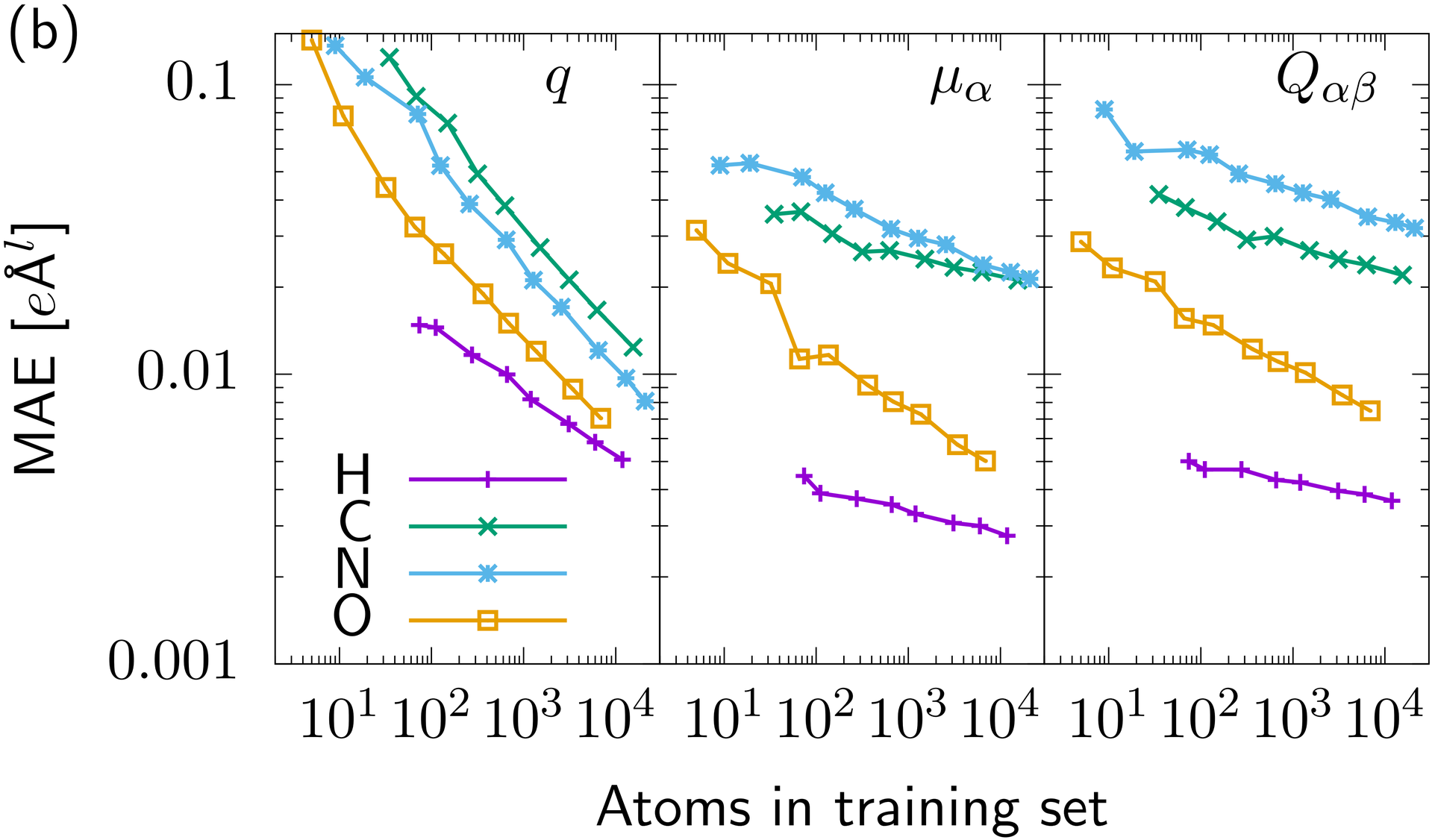
dipoles, quadrupoles rotate with the sample

$$4\pi\varepsilon_0\Phi(\mathbf{r}) = \frac{q}{R} + \frac{\mu_\alpha R_\alpha}{R^3} + \frac{1}{3}\frac{\Theta_{\alpha\beta}}{R^5}\frac{3R_\alpha R_\beta - R^2\delta_{\alpha\beta}}{} + \ldots$$

Stone, *The Theory of Intermolecular Forces*

Bereau and Meuwly, *Many-Body Effects and Electrostatics in Biomolecules*

Multipoles: Learning curves



Easier to learn H,O than C,N

Bereau, DiStasio Jr., Tkatchenko, von Lilienfeld, *JCP* **148**, 241706 (2018); JCP Editor's Choice 2018

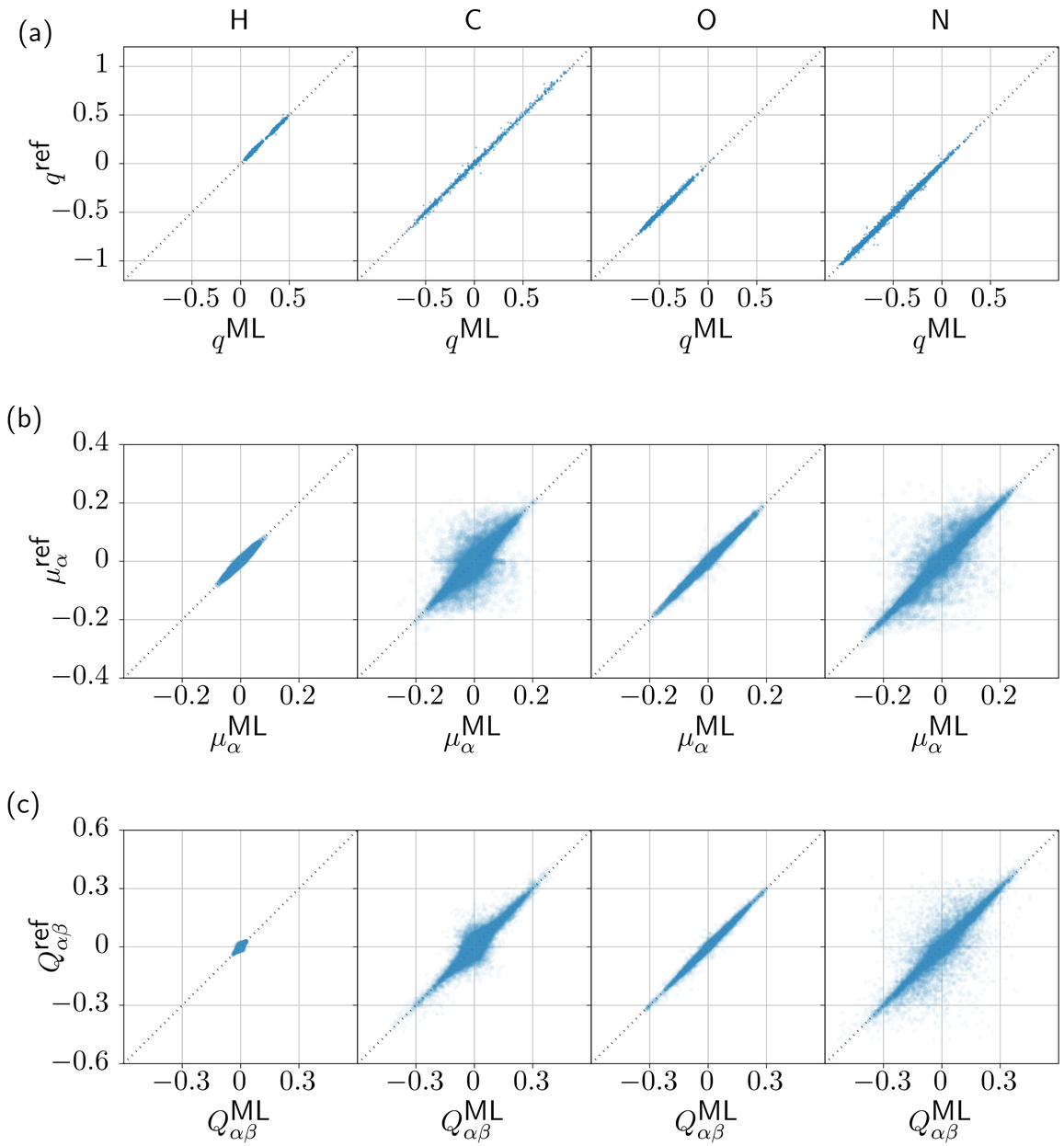
Multipoles: Correlation curves

Max-Planck-Institut
für Polymerforschung
Max Planck Institute
for Polymer Research

monopole

dipole

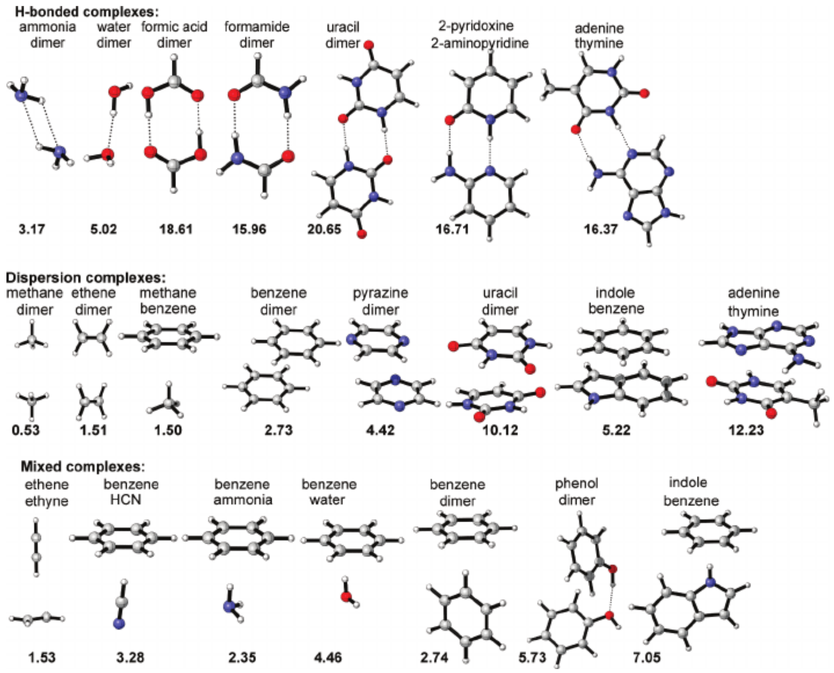
quadrupole

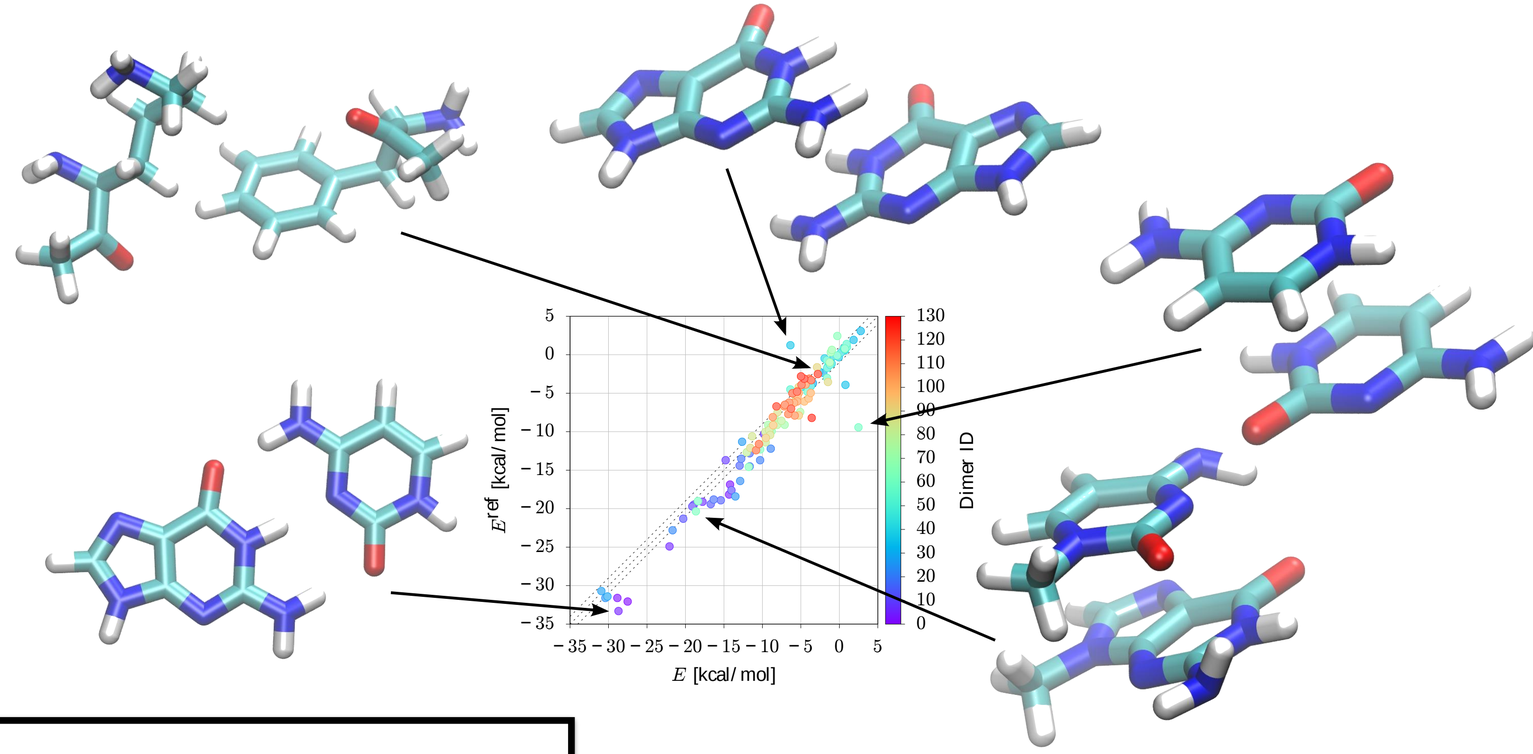


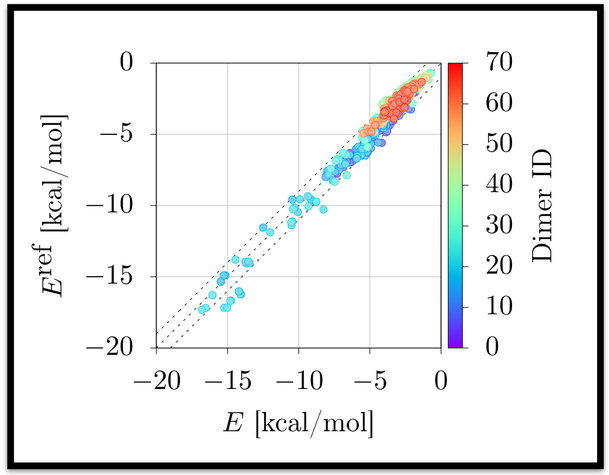
C,N have more complex valencies

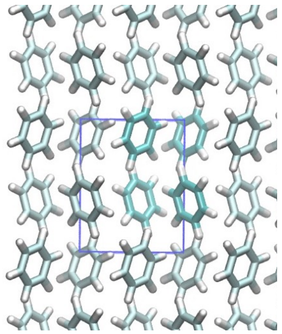
Intermolecular energy across conformations and composition

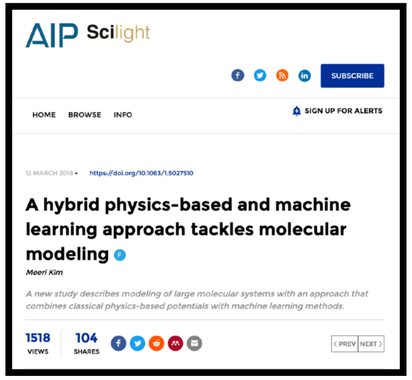
7 global parameters











Jurecka, Sponer, Cerny, Hobza, *PCCP* (2006)
Paton and Goodman, *J Chem Inf Model* (2009)

THEORY GROUP

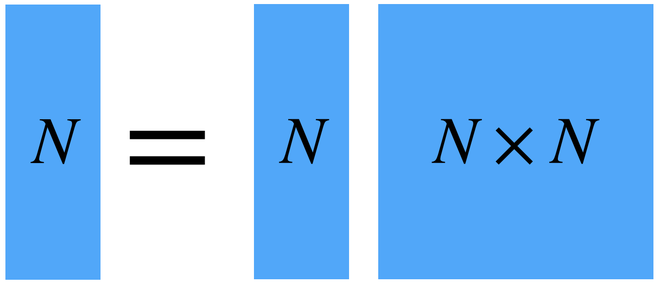
Bereau, DiStasio Jr., Tkatchenko, von Lilienfeld, *JCP* **148**, 241706 (2018); JCP Editor's Choice 2018

Energy conservation

Learning a vector field: matrix-valued kernels

Up to now: learning a *scalar* field $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$



$$\alpha_i \in \mathbb{R}$$

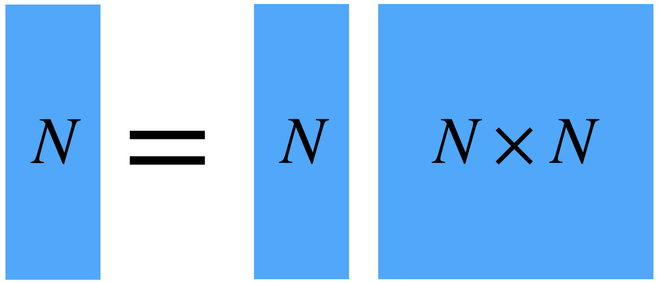
$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

Macêdo and Castro, *Learning divergence-free and curl-free vector fields with matrix-valued kernels, IMPA* (2010)
Micchelli and M. Pontil. *Neural Computation*, 17(1):177–204, 2005.

Learning a vector field: matrix-valued kernels

Up to now: learning a *scalar* field $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

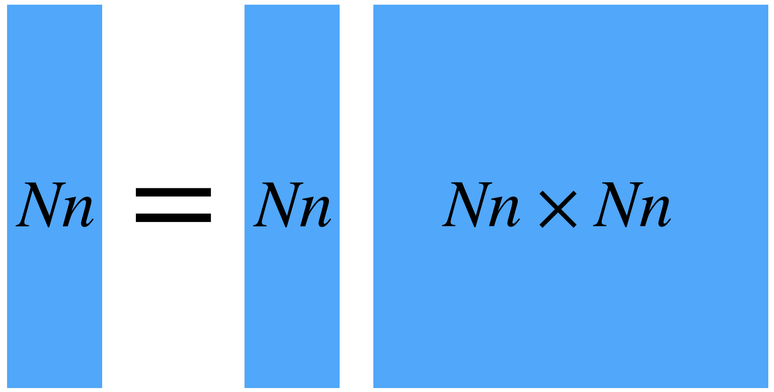


$$\alpha_i \in \mathbb{R}$$

$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

Learning a vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$



$$\alpha_i \in \mathbb{R}^n$$

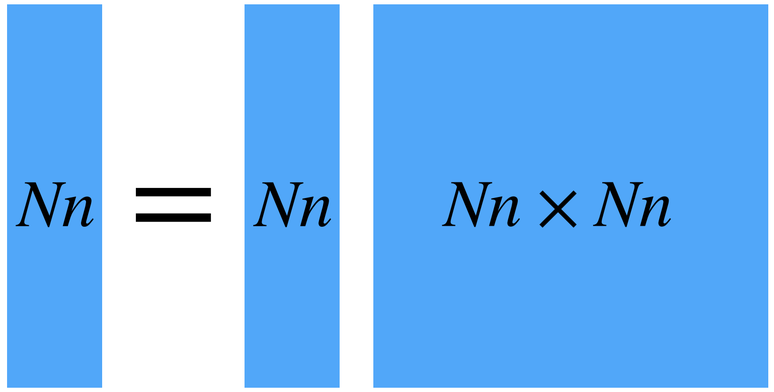
$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$$

Macêdo and Castro, *Learning divergence-free and curl-free vector fields with matrix-valued kernels, IMPA* (2010)
Micchelli and M. Pontil. *Neural Computation*, 17(1):177–204, 2005.

Matrix-valued kernel

Learning a vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$



$$\alpha_i \in \mathbb{R}^n$$

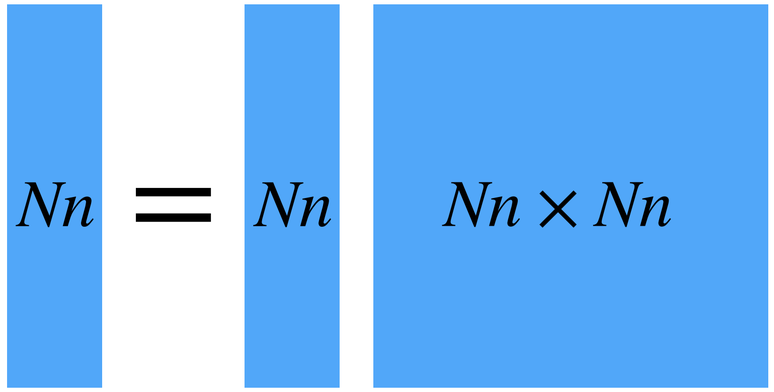
$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$$

By default: components learned *independently*. **No prior on the vector field**.

Matrix-valued kernel

Learning a vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$



$$\alpha_i \in \mathbb{R}^n$$

$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$$

By default: components learned *independently*. **No prior on the vector field**.

Time invariance leads to an **energy-conserving** force field (curl-free):

$$\nabla \times \mathbf{f} = 0$$

Design matrix-valued kernel that is *also* curl free.

Enforcing structure onto the vector field: Matrix-valued radial basis functions

Recall the (translation-invariant) kernel:

$$K(\mathbf{r}, \mathbf{r}') = \exp\left(-\frac{\|\mathbf{r}-\mathbf{r}'\|^2}{2\sigma^2}\right) = \varphi\left(\|\mathbf{r}-\mathbf{r}'\|\right)$$

scalar RBF $\Phi : \mathbb{R}^n \to \mathbb{R}$

E. Fuselier. PhD thesis, Texas A&M University (2006)

Enforcing structure onto the vector field: Matrix-valued radial basis functions

Recall the (translation-invariant) kernel:

$$K(\mathbf{r}, \mathbf{r}') = \exp\left(-\frac{\|\mathbf{r}-\mathbf{r}'\|^2}{2\sigma^2}\right) = \varphi\left(\|\mathbf{r}-\mathbf{r}'\|\right)$$

scalar RBF $\Phi : \mathbb{R}^n \to \mathbb{R}$

Construct matrix-valued RBF $\Phi : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ from a scalar RBF $\phi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$

Apply linear differential operator: $\Phi(\mathbf{x}) := (\mathscr{L}\phi)(\mathbf{x})$

THEORY GROUP

E. Fuselier. PhD thesis, Texas A&M University (2006)

Enforcing structure onto the vector field: Matrix-valued radial basis functions

Recall the (translation-invariant) kernel:

$$K(\mathbf{r}, \mathbf{r}') = \exp\left(-\frac{\|\mathbf{r}-\mathbf{r}'\|^2}{2\sigma^2}\right) = \varphi\left(\|\mathbf{r}-\mathbf{r}'\|\right)$$

scalar RBF $\Phi : \mathbb{R}^n \to \mathbb{R}$

Construct matrix-valued RBF $\Phi : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ from a scalar RBF $\phi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$

Apply linear differential operator: $\Phi(\mathbf{x}) := (\mathscr{L}\phi)(\mathbf{x})$

Example: curl-free $$(H\phi)_{ij} := \frac{\partial^2 \phi}{\partial x_i \partial x_j}$$

E. Fuselier. PhD thesis, Texas A&M University (2006)

Vector fields: curl-free and divergence-free

divergence-free

$$\Phi_{\mathrm{df}}(\mathbf{x}) = (H\phi)(\mathbf{x}) - \mathrm{Tr}\{(H\phi)(\mathbf{x})\} \cdot \mathbb{I}$$

curl-free

$$\Phi_{\mathrm{cf}}(\mathbf{x}) = -(H\phi)(\mathbf{x})$$

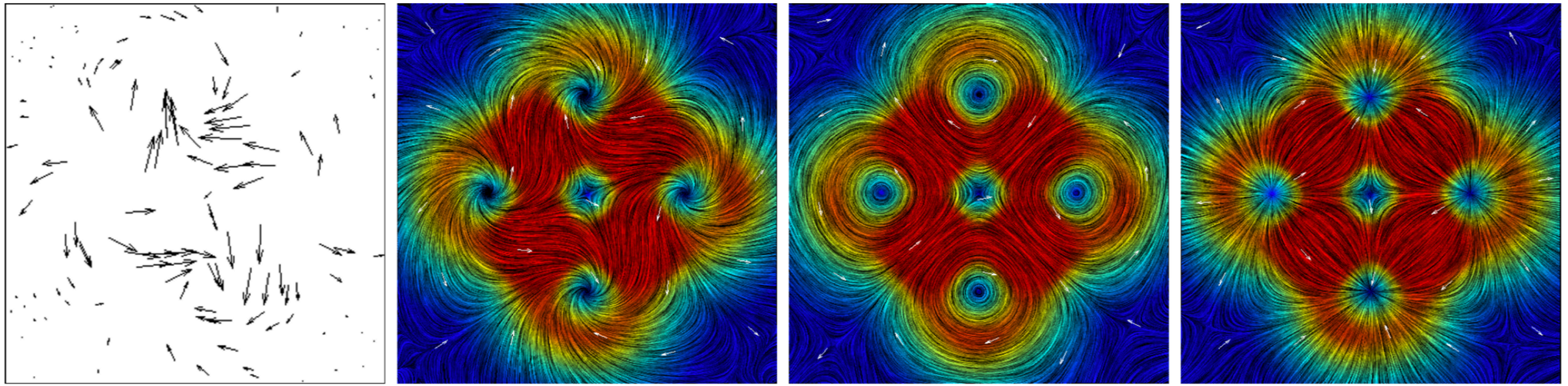
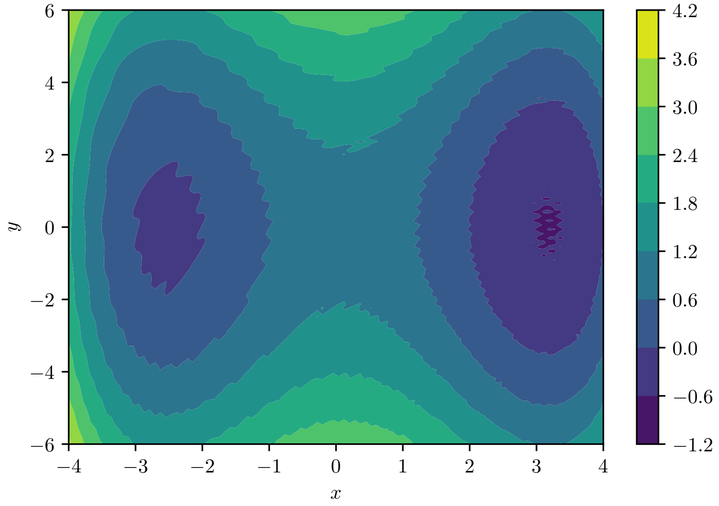


Figure 1. Learning a vector field decomposition: samples, learned field, divergence- and curl-free parts.

Macêdo and Castro, *Learning divergence-free and curl-free vector fields with matrix-valued kernels, IMPA* (2010)

Kernel learning of a 2D potential

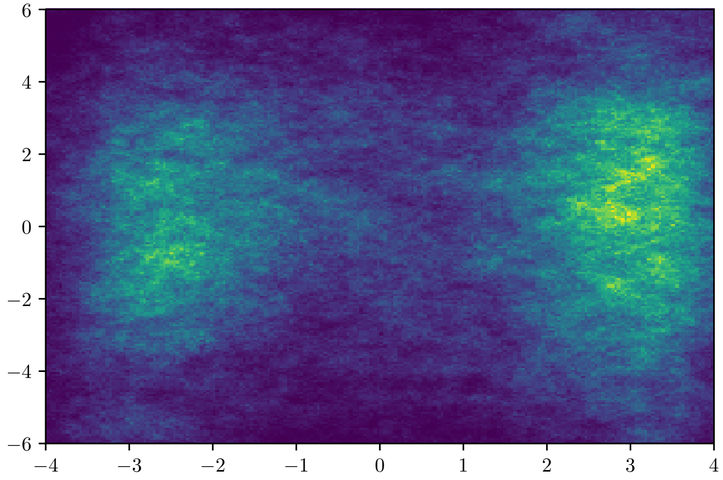
Potential



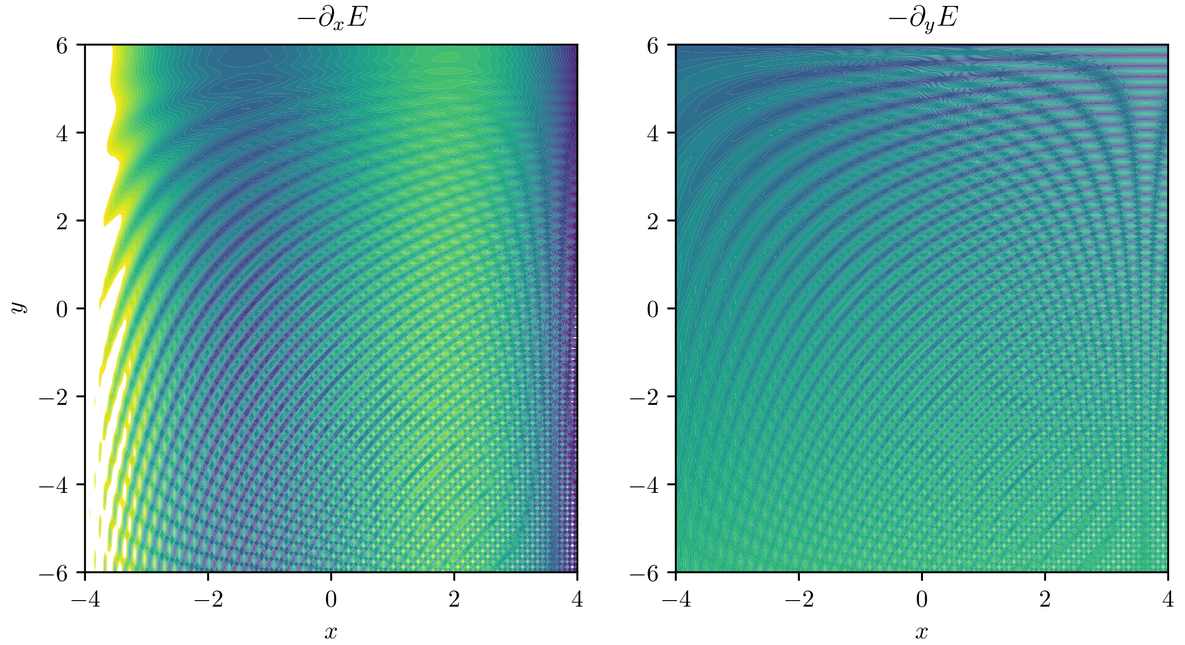
$$\frac{V(x,y)}{k_\mathrm{B}T} = \frac{1}{50}(x-4)(x-2)(x+2)(x+3) + \frac{1}{20}y^2 + \frac{1}{25}\sin(3(x+5)(y-6))$$

Potential from: Wang et al., *ACS Cent. Sci.* **5** 755–767 (2019)

Distribution (Brownian dynamics)

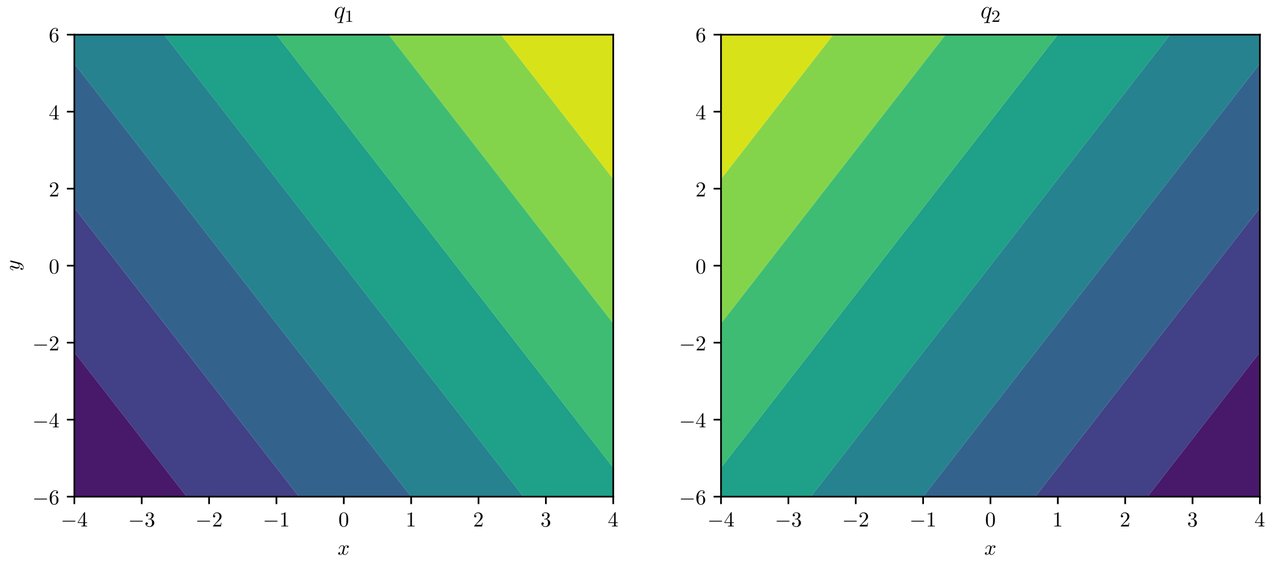


Force field



Standard kernel

$$\mathbf{q} = \begin{pmatrix} 2(x+y) \\ x-y \end{pmatrix}$$



Learn the instantaneous forces

$$(\{f^x\}, \{f^y\})$$

and assume **independence**

$$N = N \quad N \times N \qquad N = N \quad N \times N$$

$$K(\mathbf{q}, \mathbf{q}') = \exp\left(-\frac{(\mathbf{q}-\mathbf{q}')^2}{2\sigma^2}\right)$$

$$f_j^x = \sum_i \alpha_i^x \left(K(\mathbf{q}_i, \mathbf{q}_j) + \lambda \mathbb{I}\right) \qquad f_j^y = \sum_i \alpha_i^y \left(K(\mathbf{q}_i, \mathbf{q}_j) + \lambda \mathbb{I}\right)$$

Energy-conserving kernel

Learning in the gradient domain:

$$\mathrm{Cov}\left(\frac{\partial E(\mathbf{q}_i)}{\partial r^k}, \frac{\partial E(\mathbf{q}_j)}{\partial r^l}\right) = \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial^2 K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q} \partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial r_j^l} + \frac{\partial K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial r_i^k \partial r_j^l}$$

Mathias, Master Thesis, Bonn (2015); Glielmo et al., *arXiv:1905.07626; Csanyi; Tkatchenko; Müller…*

Energy-conserving kernel

Learning in the gradient domain:

$$\mathrm{Cov}\left(\frac{\partial E(\mathbf{q}_i)}{\partial r^k}, \frac{\partial E(\mathbf{q}_j)}{\partial r^l}\right) = \underbrace{\frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial^2 K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q} \partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial r_j^l}}_{\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l)} + \frac{\partial K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial r_i^k \partial r_j^l}$$

Mathias, Master Thesis, Bonn (2015); Glielmo et al., *arXiv:1905.07626; Csanyi; Tkatchenko; Müller…*

Energy-conserving kernel

Learning in the gradient domain:

$$\mathrm{Cov}\left(\frac{\partial E(\mathbf{q}_i)}{\partial r^k}, \frac{\partial E(\mathbf{q}_j)}{\partial r^l}\right) = \underbrace{\frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial^2 K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q} \partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial r_j^l}}_{\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l)} + \frac{\partial K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial r_i^k \partial r_j^l}$$

$$K(\mathbf{q}, \mathbf{q}') = \exp\left(-\frac{(\mathbf{q} - \mathbf{q}')^2}{2\sigma^2}\right)$$

$$\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l) = \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{\partial}{\partial \mathbf{q}} \frac{\partial}{\partial \mathbf{q}} \exp\left(-\frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{2\sigma^2}\right)$$

$$= \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{1}{\sigma^2} \left(1 - \frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{\sigma^2}\right) \exp\left(-\frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{2\sigma^2}\right)$$

Energy-conserving kernel

Learning in the gradient domain:

$$\mathrm{Cov}\left(\frac{\partial E(\mathbf{q}_i)}{\partial r^k}, \frac{\partial E(\mathbf{q}_j)}{\partial r^l}\right) = \underbrace{\frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial^2 K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q} \partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial r_j^l}}_{\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l)} + \cancel{\frac{\partial K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial r_i^k \partial r_j^l}}$$

$$K(\mathbf{q}, \mathbf{q}') = \exp\left(-\frac{(\mathbf{q} - \mathbf{q}')^2}{2\sigma^2}\right)$$

$$\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l) = \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{\partial}{\partial \mathbf{q}} \frac{\partial}{\partial \mathbf{q}} \exp\left(-\frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{2\sigma^2}\right)$$

$$= \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{1}{\sigma^2} \left(1 - \frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{\sigma^2}\right) \exp\left(-\frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{2\sigma^2}\right)$$

Energy-conserving kernel

Learning in the gradient domain:

$$\mathrm{Cov}\left(\frac{\partial E(\mathbf{q}_i)}{\partial r^k}, \frac{\partial E(\mathbf{q}_j)}{\partial r^l}\right) = \underbrace{\frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial^2 K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q} \partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial r_j^l}}_{\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l)} + \cancel{\frac{\partial K(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial r_i^k \partial r_j^l}}$$

$$K(\mathbf{q}, \mathbf{q}') = \exp\left(-\frac{(\mathbf{q} - \mathbf{q}')^2}{2\sigma^2}\right)$$

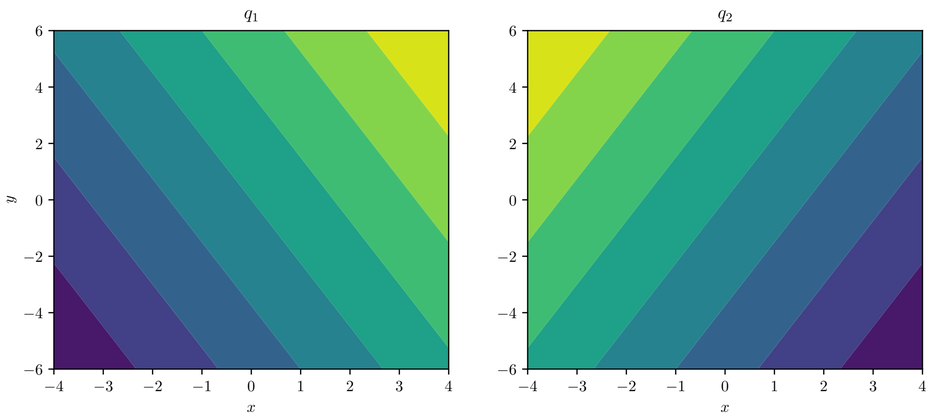
$$\mathbf{K}_{\mathrm{Hess}}(q_i^k, q_j^l) = \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{\partial}{\partial \mathbf{q}} \frac{\partial}{\partial \mathbf{q}} \exp\left(-\frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{2\sigma^2}\right)$$

$$= \boxed{\frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l}} \frac{1}{\sigma^2}\left(1 - \frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{\sigma^2}\right) \exp\left(-\frac{(\mathbf{q}_i - \mathbf{q}_j)^2}{2\sigma^2}\right)$$

enables energy conservation

Mathias, Master Thesis, Bonn (2015); Glielmo et al., *arXiv:1905.07626; Csanyi; Tkatchenko; Müller…*

Energy-conserving kernel



$$\mathbf{q} = \begin{pmatrix} 2(x+y) \\ x-y \end{pmatrix} \qquad \frac{\partial \mathbf{q}}{\partial x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad \frac{\partial \mathbf{q}}{\partial y} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$$\mathbf{q} = \begin{pmatrix} x \\ y \end{pmatrix} \qquad \frac{\partial \mathbf{q}}{\partial x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \frac{\partial \mathbf{q}}{\partial y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

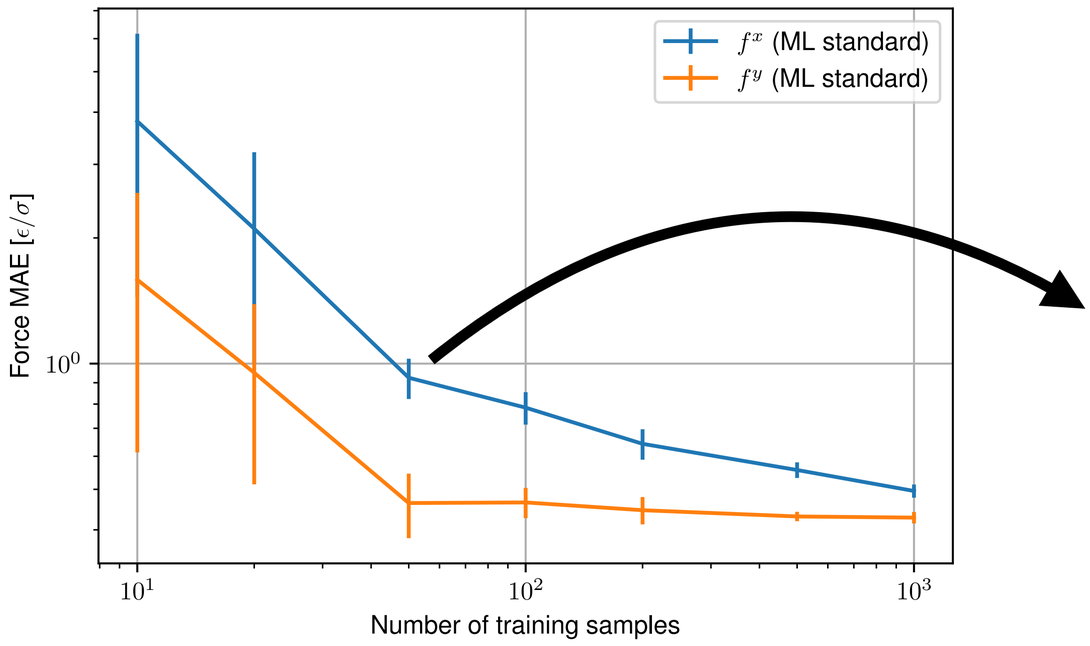
$$K(\mathbf{q}, \mathbf{q}') = \exp\left(-\frac{(\mathbf{q}-\mathbf{q}')^2}{2\sigma^2}\right)$$

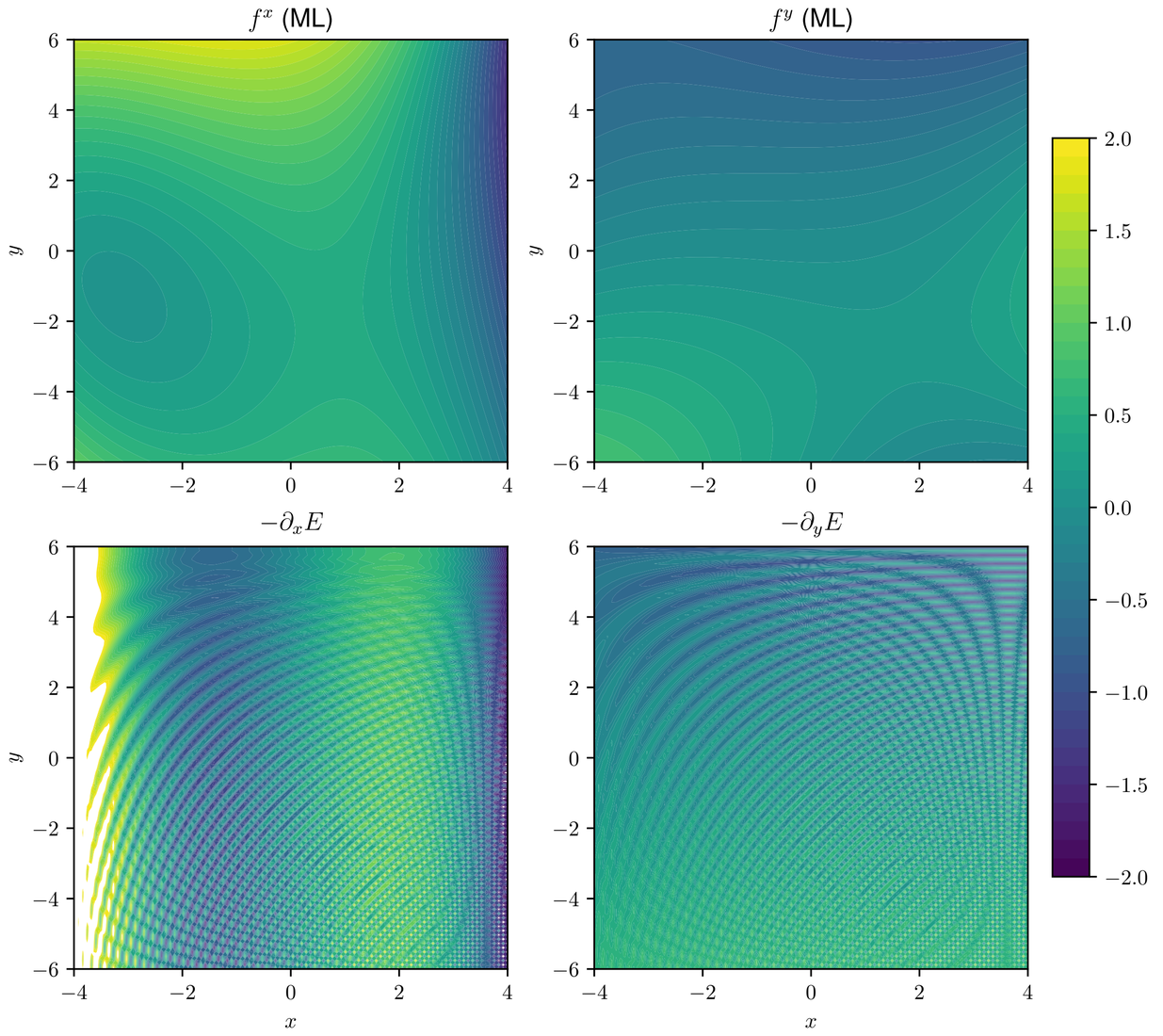
$$\mathbf{K}_{\text{Hess}}(q_i^k, q_j^l) = \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{\partial}{\partial \mathbf{q}} \frac{\partial}{\partial \mathbf{q}} \exp\left(-\frac{(\mathbf{q}_i-\mathbf{q}_j)^2}{2\sigma^2}\right)$$

$$= \frac{\partial \mathbf{q}}{\partial r_i^k} \cdot \frac{\partial \mathbf{q}}{\partial r_j^l} \frac{1}{\sigma^2}\left(1 - \frac{(\mathbf{q}_i-\mathbf{q}_j)^2}{\sigma^2}\right) \exp\left(-\frac{(\mathbf{q}_i-\mathbf{q}_j)^2}{2\sigma^2}\right)$$

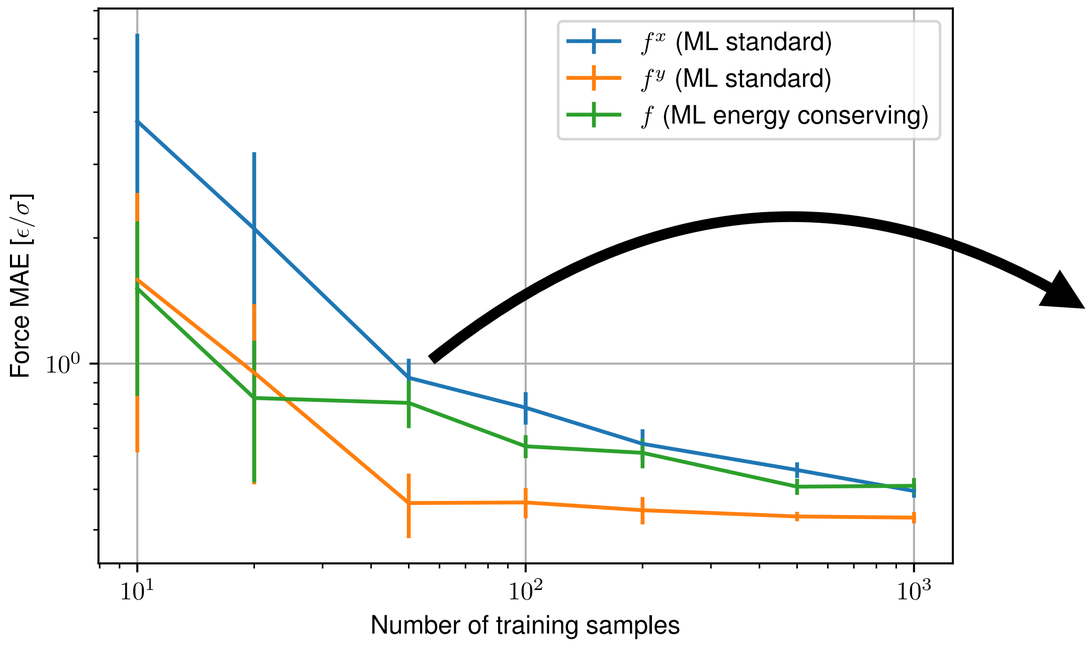
enables energy conservation

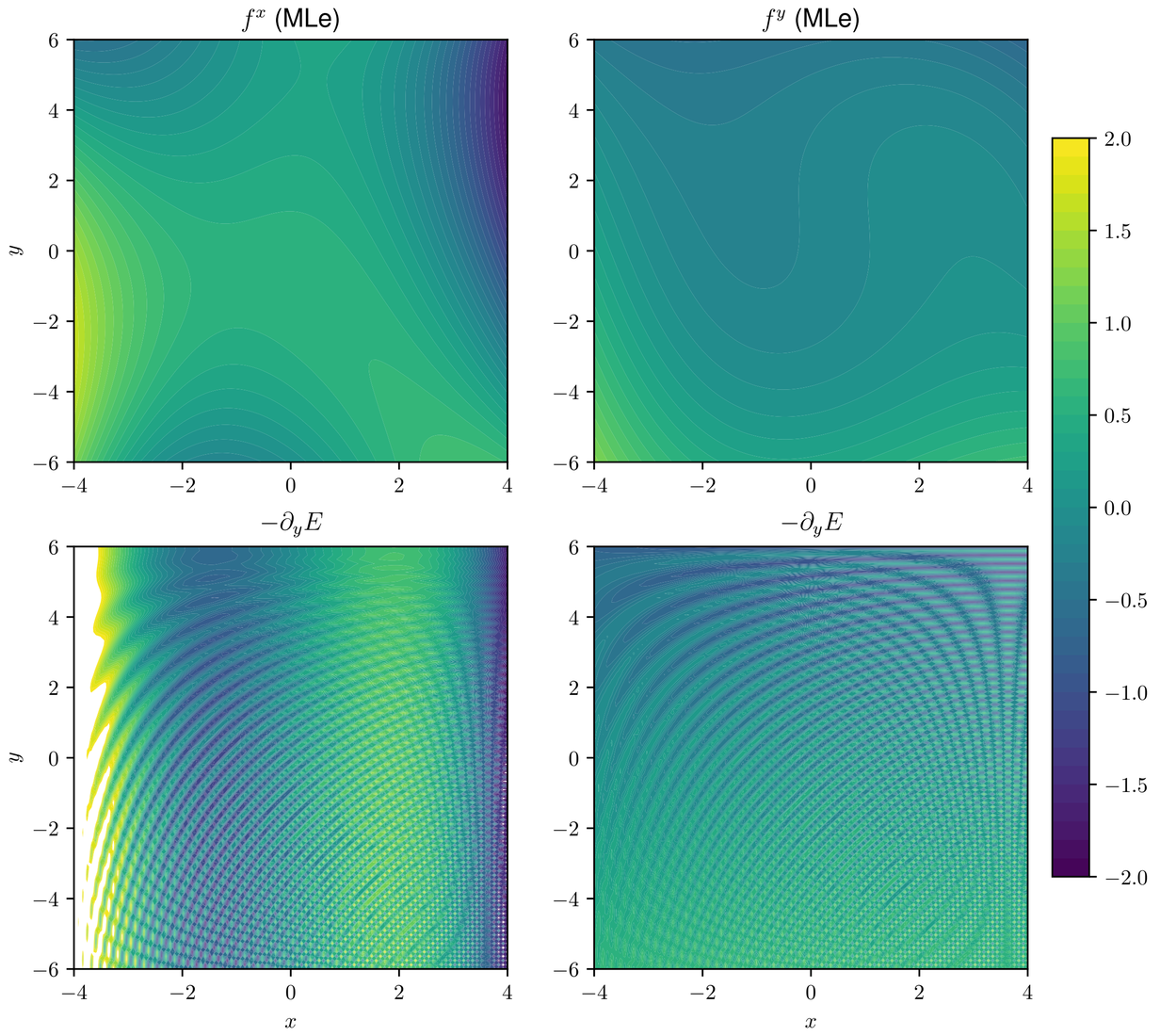
Standard kernel



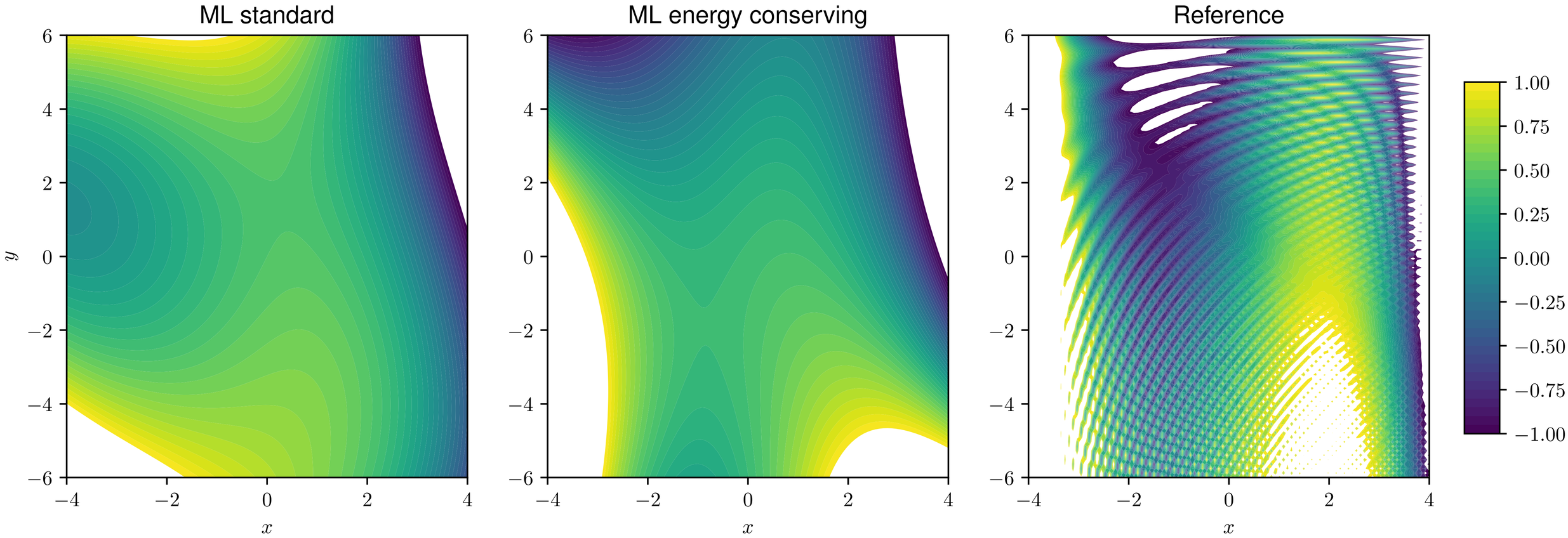


Energy-conserving ML





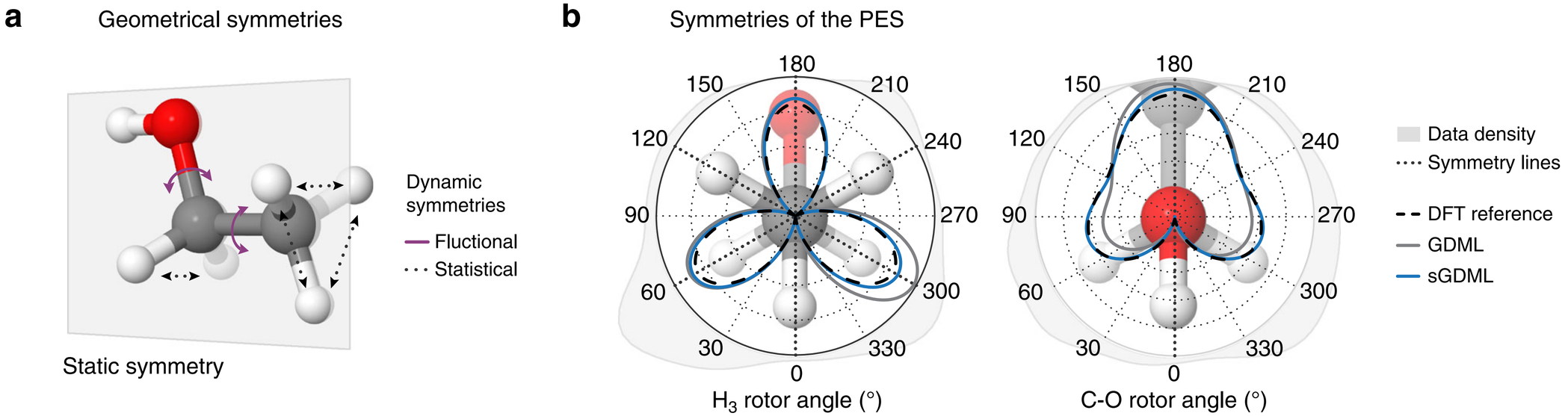
Comparison between ML models



Learning from only 50 samples

Energy-conserving force field useful in the low-data regime

Building symmetries in ML force fields



$$\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x}) = \sum_{i}^{M} \sum_{l}^{3N} \sum_{q}^{S} (\mathbf{P}_q \alpha_i)_l \frac{\partial}{\partial x_l} \nabla \kappa\left(\mathbf{x}, \mathbf{P}_q \mathbf{x}_i\right)$$

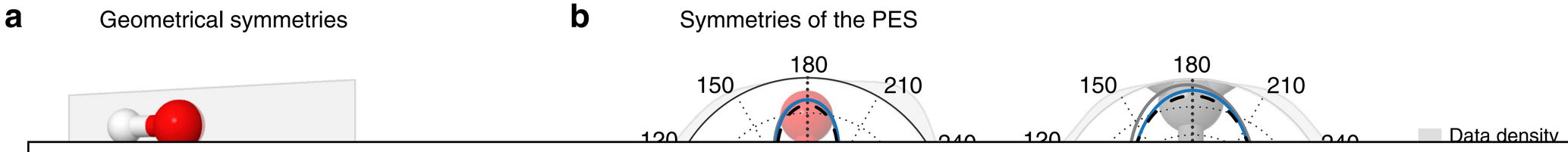
local symmetries

energy conservation

- Incorporate symmetries and conservation laws to minimize the training data
- Reduces dataset size: can target extremely accurate quantum chemistry

Chmiela *et al.*, *Nat. Comm.* **9**:3887 (2018)

Building symmetries in ML force fields



Operators in quantum machine learning: Response properties in chemical space

Cite as: J. Chem. Phys. 150, 064105 (2019); doi: 10.1063/1.5053562
Submitted: 23 August 2018 • Accepted: 19 November 2018 •
Published Online: 12 February 2019

View Online Export Citation CrossMark

Anders S. Christensen, Felix A. Faber, and O. Anatole von Lilienfeld[a)]

AFFILIATIONS

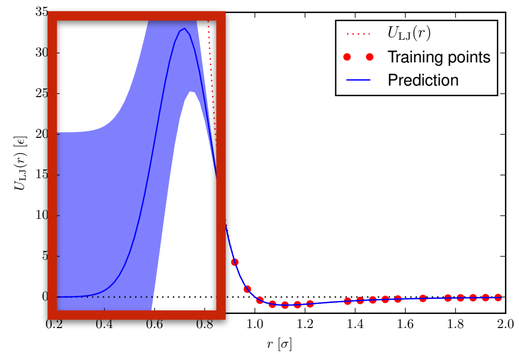
Department of Chemistry, University of Basel, Basel, Switzerland

[a)]Electronic mail: anatole.vonlilienfeld@unibas.ch

- Inc
- Reduces dataset size: can target extremely accurate quantum chemistry

Chmiela *et al.*, *Nat. Comm.* **9**:3887 (2018)

Conclusions

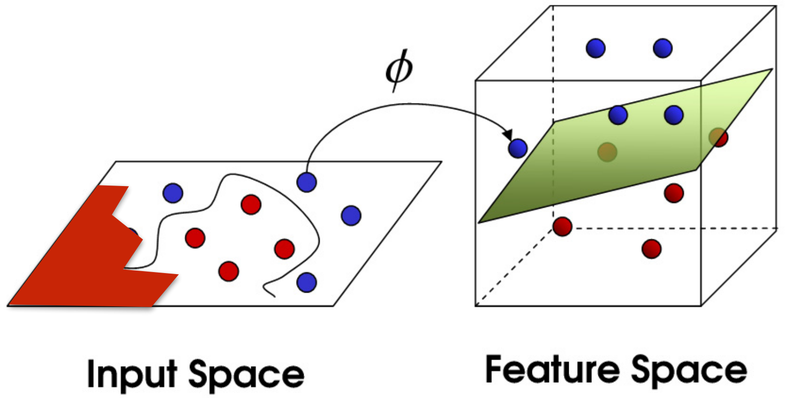


Extrapolation in ML models of energy landscapes

Can lead to catastrophic physics

Take advantage of symmetries

Noether: symmetry leads to conservation law



$$\mathbf{K}(\mathcal{S}\rho, \mathcal{S}'\rho') = \mathbf{S}\mathbf{K}(\rho, \rho')\mathbf{S}'^{\mathrm{T}}$$

Build symmetries in ML model

Work with subset of kernels that a priori satisfy conservation law