Geometric Invariants on Position-Orientation Space

with application in equivariant machine learning

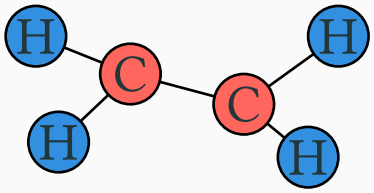
Gijs Bellaard, Bart M. N. Smets, and Remco Duits

16 April 2025

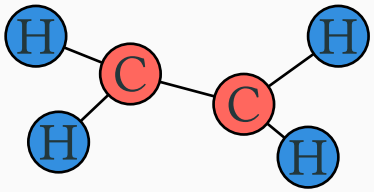
Outline

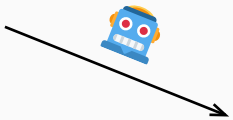
Motivation

Motivation



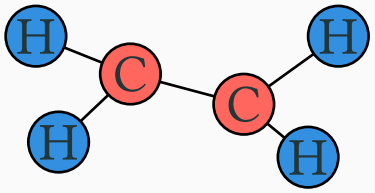
Motivation

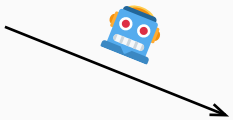




Dipole moment: 0.05 D
Isotropic polarizability: 1.55 a_0^3
Internal energy at 0K: 1.30 eV
⋮

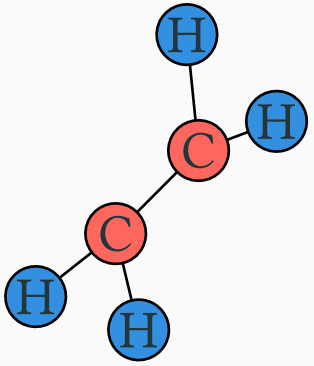
Motivation



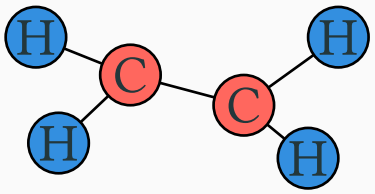


rotate
translate
reflect

Dipole moment: 0.05 D
Isotropic polarizability: 1.55 a_0^3
Internal energy at 0K: 1.30 eV
⋮

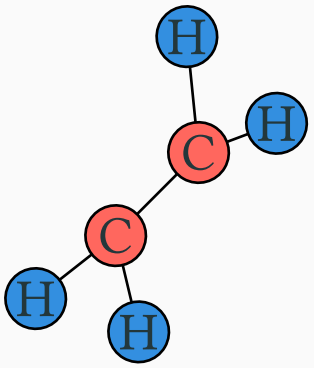


Motivation

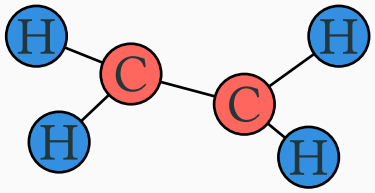


rotate
translate
reflect

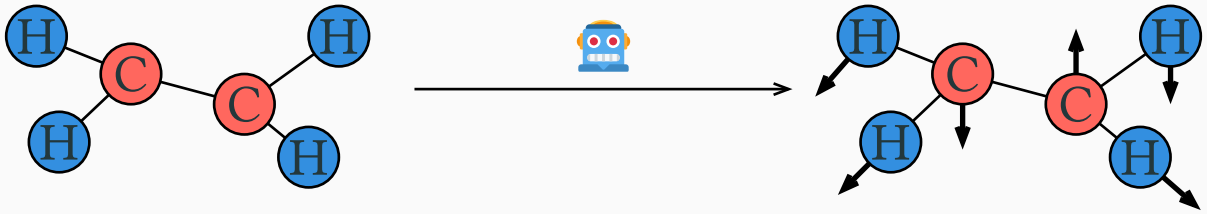
Dipole moment: 0.05 D
Isotropic polarizability: 1.55 a_0^3
Internal energy at 0K: 1.30 eV
⋮



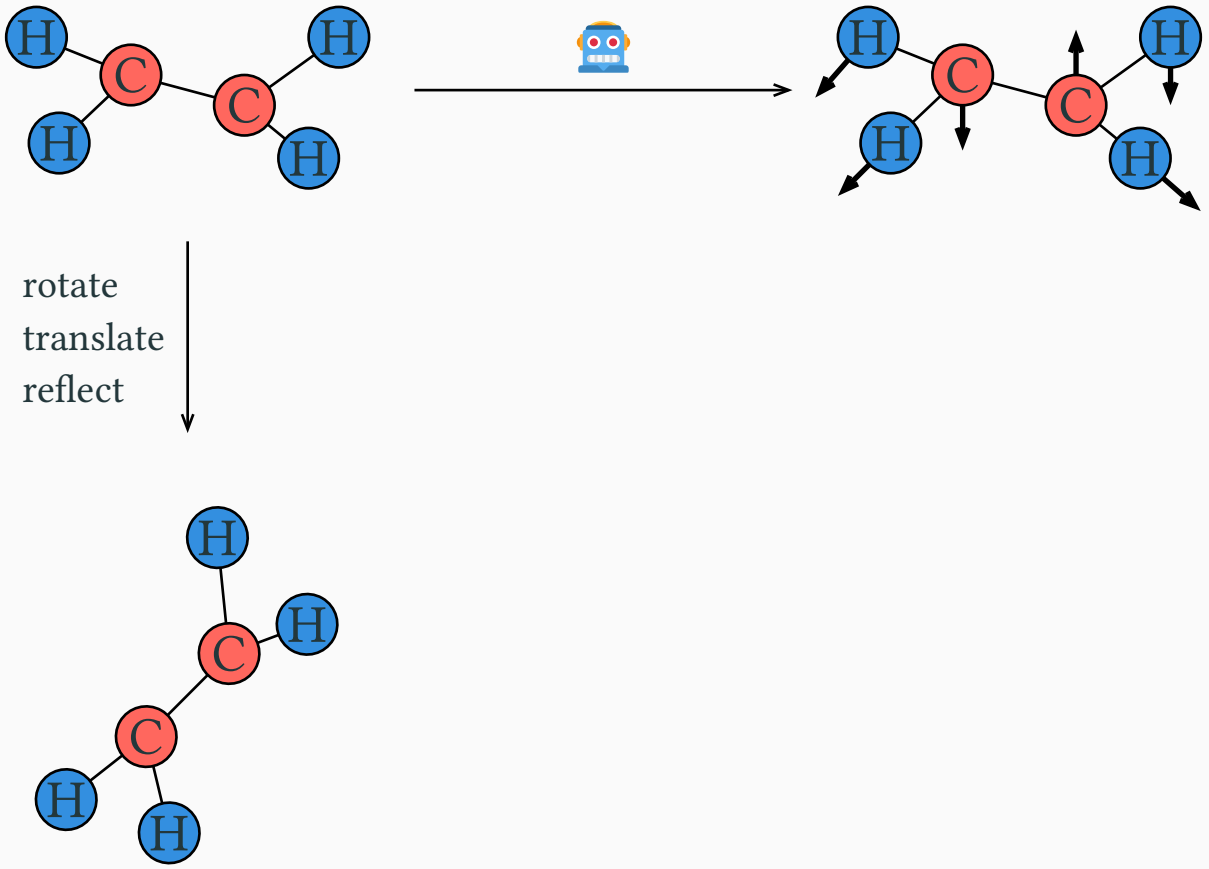
Motivation



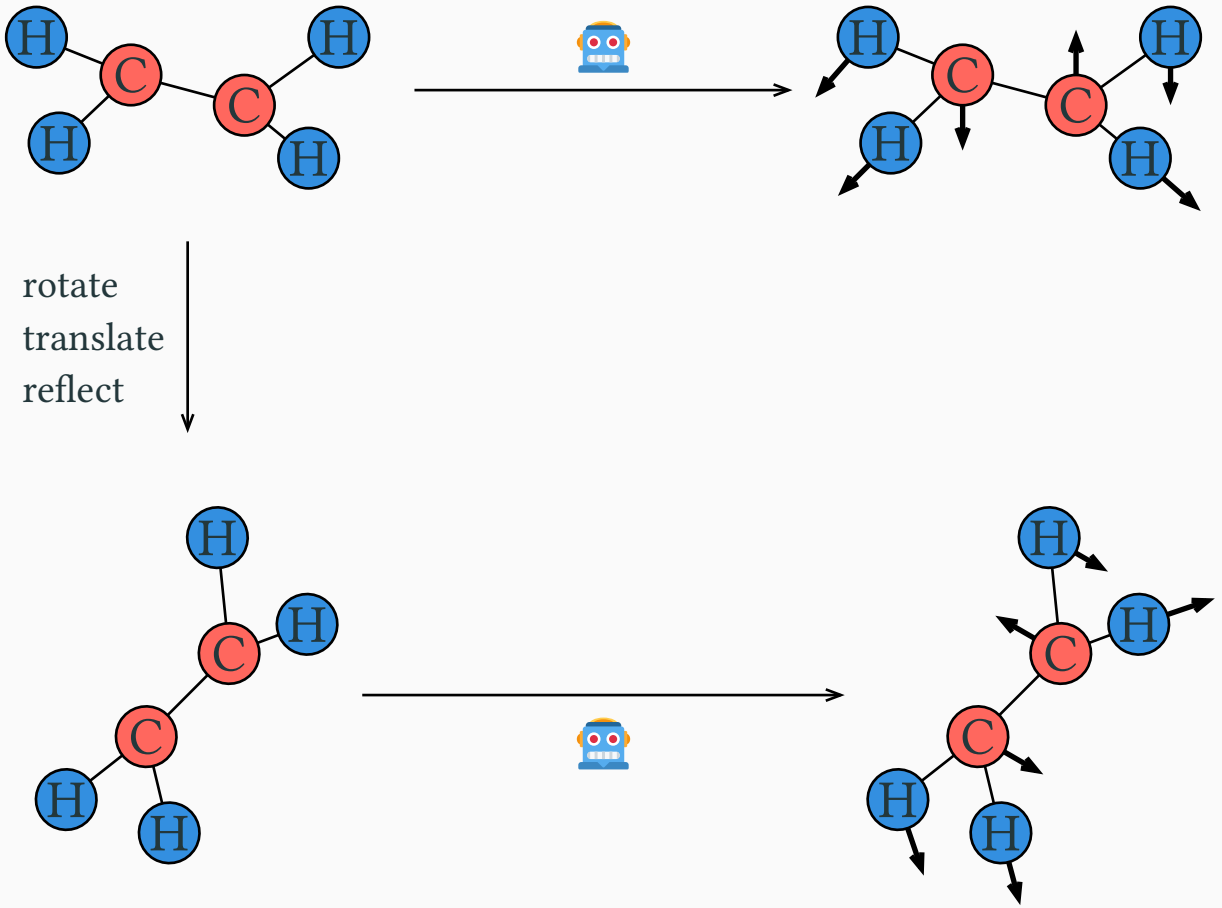
Motivation



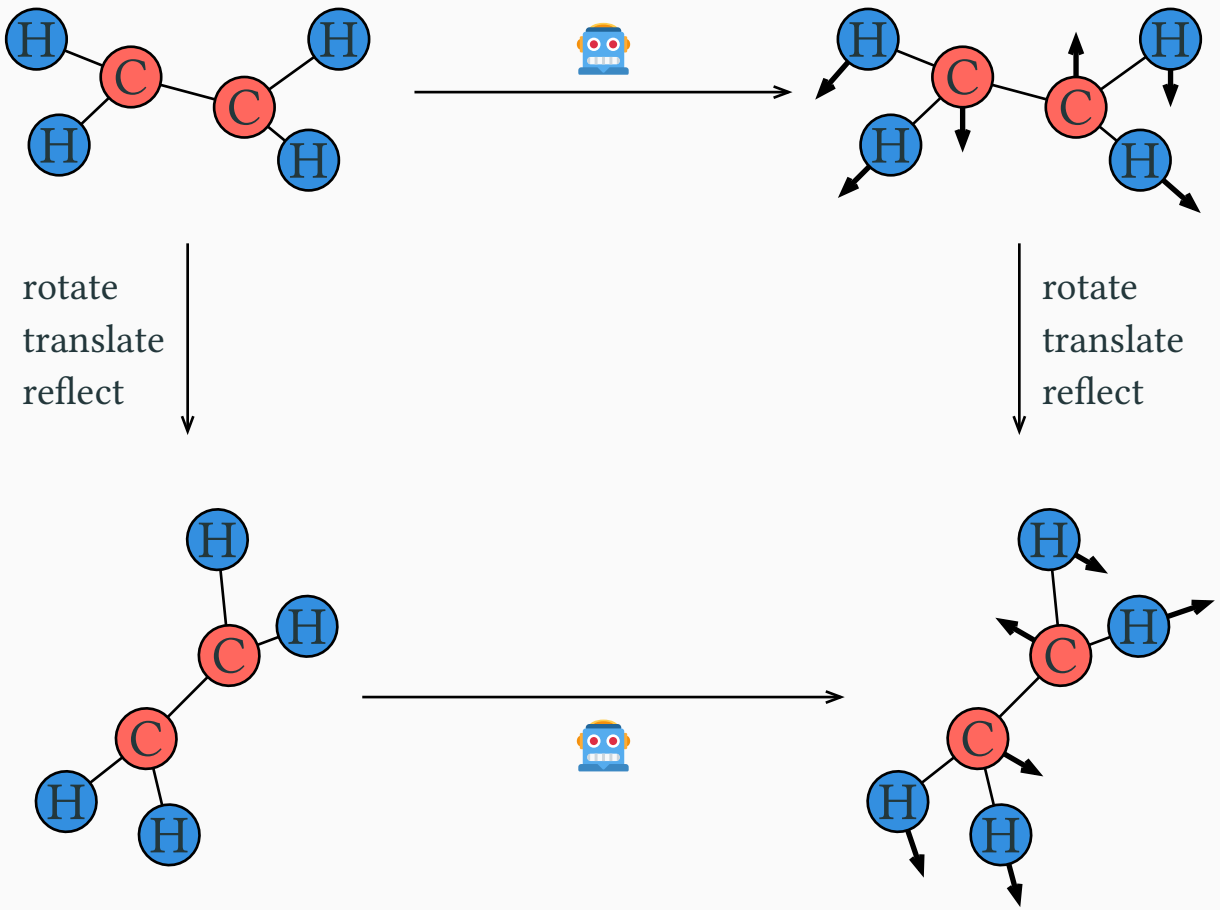
Motivation



Motivation



Motivation


H
C
C
H
H
H
H
C
C
H
H
H
rotate
translate
reflect
rotate
translate
reflect
H
C
C
H
H
H
H
C
C
H
H
H

Motivation

- GLDG does **chemistry** now?

Motivation

- GLDG does **chemistry** now?
- Actually interested in the **equivariant machine learning** part.
- We build on architectures that were benchmarked on chemical datasets so we follow suit to see if our modifications improve performance.

Motivation

- GLDG does **chemistry** now?
- Actually interested in the **equivariant machine learning** part.
- We build on architectures that were benchmarked on chemical datasets so we follow suit to see if our modifications improve performance.
- While our main applications of interest are in image processing, much of our work is very general and applies to a wide range of problems.

We should build models that **respect** translation, rotational, and reflectional transformations.

PONITA

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

- respects translation, rotational, and reflectional transformations.

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

- respects translation, rotational, and reflectional transformations.
- uses scalar fields on *position-orientation space*.

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

- respects translation, rotational, and reflectional transformations.
- uses scalar fields on *position-orientation space*.
- is faster than the typical steerable/tensor field network (Thomas et al. [2], Anderson et al. [3]).

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

- respects translation, rotational, and reflectional transformations.
- uses scalar fields on *position-orientation space*.
- is faster than the typical steerable/tensor field network (Thomas et al. [2], Anderson et al. [3]).
- does not require any representation theory of $\mathrm{SO}(3)$ (no Clebsch-Gordan coefficients, Wigner D-matrices, etc...).

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

- respects translation, rotational, and reflectional transformations.
- uses scalar fields on *position-orientation space*.
- is faster than the typical steerable/tensor field network (Thomas et al. [2], Anderson et al. [3]).
- does not require any representation theory of $\mathrm{SO}(3)$ (no Clebsch-Gordan coefficients, Wigner D-matrices, etc...).
- achieves state-of-the-art accuracy on predicting molecular dynamics & properties.

PONITA

PONITA 🐴 is a neural network architecture by Bekkers et al. [1] that...

- respects translation, rotational, and reflectional transformations.
- uses scalar fields on *position-orientation space*.
- is faster than the typical steerable/tensor field network (Thomas et al. [2], Anderson et al. [3]).
- does not require any representation theory of $\mathrm{SO}(3)$ (no Clebsch-Gordan coefficients, Wigner D-matrices, etc...).
- achieves state-of-the-art accuracy on predicting molecular dynamics & properties.

Euclidean Group $\mathrm{E}(3)$ & Position-Orientation Space $\mathbb{M}_3$

Euclidean Group $\mathrm{E}(3)$ & Position-Orientation Space $\mathbb{M}_3$

Definition: The *Euclidean group* or *rigid transformation group* in 3 dimensions is

$$\mathrm{E}(3) := \{(t, Q) \in \mathbb{R}^3 \times \mathbb{R}^{3\times 3} \mid Q^\top Q = I\}$$

that is, the group of all translations, rotations, and reflections.

Euclidean Group $\mathrm{E}(3)$ & Position-Orientation Space $\mathbb{M}_3$

Definition: The *Euclidean group* or *rigid transformation group* in 3 dimensions is

$$\mathrm{E}(3) := \{(t, Q) \in \mathbb{R}^3 \times \mathbb{R}^{3\times 3} \mid Q^\top Q = I\}$$

that is, the group of all translations, rotations, and reflections.

Definition: The space of *three-dimensional position-orientations* is:

$$\mathbb{M}_3 := \{(x, n) \in \mathbb{R}^3 \times \mathbb{R}^3 \mid \|n\| = 1\}$$

Euclidean Group $\mathrm{E}(3)$ & Position-Orientation Space $\mathbb{M}_3$

Definition: The *Euclidean group* or *rigid transformation group* in 3 dimensions is

$$\mathrm{E}(3) := \{(t, Q) \in \mathbb{R}^3 \times \mathbb{R}^{3\times 3} \mid Q^\top Q = I\}$$

that is, the group of all translations, rotations, and reflections.

Definition: The space of *three-dimensional position-orientations* is:

$$\mathbb{M}_3 := \{(x, n) \in \mathbb{R}^3 \times \mathbb{R}^3 \mid \|n\| = 1\}$$

Definition: We define the action $\triangleright : \mathrm{E}(3) \times \mathbb{M}_3 \to \mathbb{M}_3$

$$(t, Q) \triangleright (x, n) = (t + Qx, Qn)$$

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- In the PONITA architecture all data (feature maps) are stored as scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$ on position-orientation space $\mathbb{M}_3$.

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- In the PONITA architecture all data (feature maps) are stored as scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$ on position-orientation space $\mathbb{M}_3$.
- Consider a linear operator Φ to process such a field f, as is common in neural networks:

$$(\Phi f)(p) := \int_{\mathbb{M}_3} k(p, q) f(q) \, \mathrm{d}q,$$

with kernel $k : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- In the PONITA architecture all data (feature maps) are stored as scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$ on position-orientation space $\mathbb{M}_3$.
- Consider a linear operator Φ to process such a field f, as is common in neural networks:

$$(\Phi f)(p) := \int_{\mathbb{M}_3} k(p, q) f(q) \, \mathrm{d}q,$$

  with kernel $k : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.
- To make the operator Φ *equivariant* it suffices to make the kernel *invariant*:

$$k(g \rhd p, g \rhd q) = k(p, q) \quad \text{for all} \quad p, q \in \mathbb{M}_3 \quad \text{and} \quad g \in \mathrm{E}(3)$$

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- In the PONITA architecture all data (feature maps) are stored as scalar fields $f : \mathbb{M}_3 \to \mathbb{R}$ on position-orientation space $\mathbb{M}_3$.
- Consider a linear operator Φ to process such a field f, as is common in neural networks:

$$(\Phi f)(p) := \int_{\mathbb{M}_3} k(p, q) f(q) \, \mathrm{d}q,$$

  with kernel $k : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.
- To make the operator Φ *equivariant* it suffices to make the kernel *invariant*:

$$k(g \rhd p, g \rhd q) = k(p, q) \quad \text{for all} \quad p, q \in \mathbb{M}_3 \quad \text{and} \quad g \in \mathrm{E}(3)$$

- So, to make PONITA respect certain transformations we are motivated to study **invariants** $\iota : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$, that being functions with the property that

$$\iota(g \rhd p, g \rhd q) = \iota(p, q) \quad \text{for all} \quad p, q \in \mathbb{M}_3 \quad \text{and} \quad g \in \mathrm{E}(3)$$

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Consider any collection of invariants $\iota_1, \ldots, \iota_n : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Consider any collection of invariants $\iota_1, ..., \iota_n : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.
- We can create a new invariant ι' easily by considering any $h : \mathbb{R}^n \to \mathbb{R}$ and defining $\iota' = h(\iota_1, ..., \iota_n)$.

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Consider any collection of invariants $\iota_1, ..., \iota_n : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.
- We can create a new invariant ι' easily by considering any $h : \mathbb{R}^n \to \mathbb{R}$ and defining $\iota' = h(\iota_1, ..., \iota_n)$.
- This observation has an immediate application in PONITA: we can decide to parameterize the kernels k by, for example, a multi-layer perceptron $\text{MLP}_\theta : \mathbb{R}^n \to \mathbb{R}$ with (trainable) parameters θ, and plugging in a predesigned collection of n invariants: $k = \text{MLP}_\theta(\iota_1, ..., \iota_n)$.

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Consider any collection of invariants $\iota_1, ..., \iota_n : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}$.
- We can create a new invariant ι' easily by considering any $h : \mathbb{R}^n \to \mathbb{R}$ and defining $\iota' = h(\iota_1, ..., \iota_n)$.
- This observation has an immediate application in PONITA: we can decide to parameterize the kernels k by, for example, a multi-layer perceptron $\text{MLP}_\theta : \mathbb{R}^n \to \mathbb{R}$ with (trainable) parameters θ, and plugging in a predesigned collection of n invariants: $k = \text{MLP}_\theta(\iota_1, ..., \iota_n)$.
- This motivates looking into what an **"optimal"** collection of invariants would be, so that we can construct networks that are as expressive and efficient as possible.

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Suppose we have a collection of invariants where one of them is a function of the others. If this happens we say the collection of invariants is **dependent**.

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Suppose we have a collection of invariants where one of them is a function of the others. If this happens we say the collection of invariants is **dependent**.
- A dependent collection is *not* "optimal" in the sense that we could remove the dependent invariant and lose no expressiveness.

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Suppose we have a collection of invariants where one of them is a function of the others. If this happens we say the collection of invariants is **dependent**.
- A dependent collection is *not* "optimal" in the sense that we could remove the dependent invariant and lose no expressiveness.
- Suppose we have a collection of invariants for which we know that *any* other invariant one can think of is a function of them. We say such a collection of invariants is **universal**.

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

- Suppose we have a collection of invariants where one of them is a function of the others. If this happens we say the collection of invariants is **dependent**.
- A dependent collection is *not* "optimal" in the sense that we could remove the dependent invariant and lose no expressiveness.
- Suppose we have a collection of invariants for which we know that *any* other invariant one can think of is a function of them. We say such a collection of invariants is **universal**.
- A universal collection of invariants is "optimal" in the sense that there is no reason to add another invariant because we gain no expressiveness.

Can we find a collection of $\mathrm{E}(3)$ invariants on $\mathbb{M}_3 \times \mathbb{M}_3$ that is both **independent** and **universal**?

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

Definition (Original PONITA Invariants): Write $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$. Bekkers et al. [1] propose the following collection of three invariants:

$$\iota_1(p_1, p_2) = (x_2 - x_1) \cdot n_1$$
$$\iota_2(p_1, p_2) = \|(x_2 - x_1) - \iota_1 n_1\|$$
$$\iota_3(p_1, p_2) = n_1 \cdot n_2$$

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

Definition (Original PONITA Invariants): Write $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$. Bekkers et al. [1] propose the following collection of three invariants:

$$\iota_1(p_1, p_2) = (x_2 - x_1) \cdot n_1$$
$$\iota_2(p_1, p_2) = \|(x_2 - x_1) - \iota_1 n_1\|$$
$$\iota_3(p_1, p_2) = n_1 \cdot n_2$$

Proposition: The original invariants are independent but *not* universal.

$\mathrm{E}(3)$ Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

Definition (Our Invariants): Write $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$. We propose the following collection of four invariants:

$$\begin{aligned}
\iota_1(p_1, p_2) &= (x_2 - x_1) \cdot n_1 \\
\iota_2(p_1, p_2) &= (x_2 - x_1) \cdot n_2 \\
\iota_3(p_1, p_2) &= (x_2 - x_1) \cdot (x_2 - x_1) \\
\iota_4(p_1, p_2) &= n_1 \cdot n_2
\end{aligned}$$

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

Definition (Our Invariants): Write $p_1 = (x_1, n_1), p_2 = (x_2, n_2) \in \mathbb{M}_3$. We propose the following collection of four invariants:

$$\iota_1(p_1, p_2) = (x_2 - x_1) \cdot n_1$$
$$\iota_2(p_1, p_2) = (x_2 - x_1) \cdot n_2$$
$$\iota_3(p_1, p_2) = (x_2 - x_1) \cdot (x_2 - x_1)$$
$$\iota_4(p_1, p_2) = n_1 \cdot n_2$$

Theorem: In Bellaard et al. [4] is it shown our invariants are universal and independent.

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

Experiment

- QM9: predict chemical properties of small organic molecules from their graph.
- PONITA: original invariants (3) versus our universal set (4).

E(3) Invariants on $\mathbb{M}_3 \times \mathbb{M}_3$

Target	Unit	Original	Universal (Ours)	Difference %
μ	D	0.0195	0.0166	−15.0
α	a_0^3	0.0557	0.0489	−12.1
$\varepsilon_{\text{homo}}$	eV	0.0226	0.0202	−10.4
$\varepsilon_{\text{lumo}}$	eV	0.0206	0.0187	−9.0
$\Delta\varepsilon$	eV	0.0415	0.0378	−8.9
$\langle R^2 \rangle$	a_0^2	0.4160	0.4251	+2.2
ZPVE	meV	1.5647	1.5241	−2.6
U_0	eV	0.9920	1.0285	+3.7
U	eV	1.3593	0.7362	−45.8
H	eV	1.0205	0.6934	−32.1
G	eV	1.1856	0.7721	−34.9
c_v	cal/mol·K	0.0292	0.0270	−7.4

PONITA trained to predict chemical properties of various molecules (QM9 dataset [5], [6]). Mean absolute error on the test set is reported (lower is better). Our universal invariants perform better.

Using a **universal** set of invariants has a **significant positive** impact on the accuracy of the PONITA model when predicting molecular properties.

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

- $\mathrm{E}(3)$ invariant *Riemannian metrics* $\mathcal{G}$ on $\mathbb{M}_3$ appear in various works related to enhancement and denoising of MRI Data (Portegies et al. [7], Duits et al. [8] Smets et al. [9]).

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

- $\mathrm{E}(3)$ invariant *Riemannian metrics* $\mathcal{G}$ on $\mathbb{M}_3$ appear in various works related to enhancement and denoising of MRI Data (Portegies et al. [7], Duits et al. [8] Smets et al. [9]).
- Such invariant metrics $\mathcal{G}$ are used to define (nonlinear) PDEs on $\mathbb{M}_3$ which then process the data $f : \mathbb{M}_3 \to \mathbb{R}$ in a $\mathrm{E}(3)$ equivariant manner, such as...

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

- $\mathrm{E}(3)$ invariant *Riemannian metrics* $\mathcal{G}$ on $\mathbb{M}_3$ appear in various works related to enhancement and denoising of MRI Data (Portegies et al. [7], Duits et al. [8] Smets et al. [9]).
- Such invariant metrics $\mathcal{G}$ are used to define (nonlinear) PDEs on $\mathbb{M}_3$ which then process the data $f : \mathbb{M}_3 \to \mathbb{R}$ in a $\mathrm{E}(3)$ equivariant manner, such as...

$$\text{Diffusion} \quad \frac{\partial f}{\partial t} = \frac{1}{2}\Delta f$$

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

- $\mathrm{E}(3)$ invariant *Riemannian metrics* $\mathcal{G}$ on $\mathbb{M}_3$ appear in various works related to enhancement and denoising of MRI Data (Portegies et al. [7], Duits et al. [8] Smets et al. [9]).
- Such invariant metrics $\mathcal{G}$ are used to define (nonlinear) PDEs on $\mathbb{M}_3$ which then process the data $f : \mathbb{M}_3 \to \mathbb{R}$ in a $\mathrm{E}(3)$ equivariant manner, such as...

$$\text{Diffusion} \quad \frac{\partial f}{\partial t} = \frac{1}{2}\Delta f$$

$$\text{Dilation/Erosion} \quad \frac{\partial f}{\partial t} = \pm\frac{1}{2}\|\nabla f\|^2$$

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

- $\mathrm{E}(3)$ invariant *Riemannian metrics* $\mathcal{G}$ on $\mathbb{M}_3$ appear in various works related to enhancement and denoising of MRI Data (Portegies et al. [7], Duits et al. [8] Smets et al. [9]).
- Such invariant metrics $\mathcal{G}$ are used to define (nonlinear) PDEs on $\mathbb{M}_3$ which then process the data $f : \mathbb{M}_3 \to \mathbb{R}$ in a $\mathrm{E}(3)$ equivariant manner, such as...

$$\text{Diffusion} \quad \frac{\partial f}{\partial t} = \frac{1}{2}\Delta f$$

$$\text{Dilation/Erosion} \quad \frac{\partial f}{\partial t} = \pm\frac{1}{2}\|\nabla f\|^2$$

$$\text{Total Variation Flow} \quad \frac{\partial f}{\partial t} = \nabla \cdot \frac{\nabla f}{\|\nabla f\|}$$

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

- $\mathrm{E}(3)$ invariant *Riemannian metrics* $\mathcal{G}$ on $\mathbb{M}_3$ appear in various works related to enhancement and denoising of MRI Data (Portegies et al. [7], Duits et al. [8] Smets et al. [9]).
- Such invariant metrics $\mathcal{G}$ are used to define (nonlinear) PDEs on $\mathbb{M}_3$ which then process the data $f : \mathbb{M}_3 \to \mathbb{R}$ in a $\mathrm{E}(3)$ equivariant manner, such as...

$$\text{Diffusion} \quad \frac{\partial f}{\partial t} = \frac{1}{2}\Delta f$$

$$\text{Dilation/Erosion} \quad \frac{\partial f}{\partial t} = \pm\frac{1}{2}\|\nabla f\|^2$$

$$\text{Total Variation Flow} \quad \frac{\partial f}{\partial t} = \nabla \cdot \frac{\nabla f}{\|\nabla f\|}$$

...where gradient ∇, norm $\|\cdot\|$, Laplacian Δ, and divergence $\nabla\cdot$, *all* depend on $\mathcal{G}$.

Can we **classify all** $\mathrm{E}(3)$ invariant metrics on $\mathbb{M}_3$?

$\mathrm{E}(3)$ Invariant Metrics on $\mathbb{M}_3$

Theorem: In Bellaard et al. [10] it is shown that every $\mathrm{E}(3)$ invariant Riemannian metric tensor field $\mathcal{G}$ on $\mathbb{M}_3$ yields a norm of the form

$$\|(p, \dot{p})\|_{\mathcal{G}}^2 = (w_1 \, |\dot{x} \cdot n|)^2 + (w_2 \, \|\dot{x} \times n\|)^2 + (w_3 \, \|\dot{n}\|)^2 + w_4 \, \dot{x} \cdot \dot{n}$$

where $p = (x, n) \in \mathbb{M}_3$, $\dot{p} = (\dot{x}, \dot{n}) \in T_p\mathbb{M}_3$, and $w_i \in \mathbb{R}$ constants called the *metric parameters*.

Mav Distance

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f$$

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f \qquad \longrightarrow \quad f_t(p) \approx \int_{\mathbb{M}_3} \exp\left(-\frac{d(p,q)^2}{2t}\right) \cdot f_0(q)\,\mathrm{d}q$$

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f \qquad \longrightarrow \quad f_t(p) \approx \int_{\mathbb{M}_3} \exp\left(-\frac{d(p,q)^2}{2t}\right) \cdot f_0(q)\,\mathrm{d}q$$

$$\frac{\partial f}{\partial t} = -\frac{1}{2}\|\nabla f\|^2$$

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f \quad \longrightarrow \quad f_t(p) \approx \int_{\mathbb{M}_3} \exp\left(-\frac{d(p,q)^2}{2t}\right) \cdot f_0(q)\,\mathrm{d}q$$

$$\frac{\partial f}{\partial t} = -\frac{1}{2}\|\nabla f\|^2 \quad \longrightarrow \quad f_t(p) = \inf_{q \in \mathbb{M}_3} \frac{d(p,q)^2}{2t} + f_0(q)$$

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f \quad \longrightarrow \quad f_t(p) \approx \int_{\mathbb{M}_3} \exp\left(-\frac{d(p,q)^2}{2t}\right) \cdot f_0(q)\,\mathrm{d}q$$

$$\frac{\partial f}{\partial t} = -\frac{1}{2}\|\nabla f\|^2 \quad \longrightarrow \quad f_t(p) = \inf_{q \in \mathbb{M}_3} \frac{d(p,q)^2}{2t} + f_0(q)$$

- However, the Riemannian distance d, in general, is **expensive** to compute.

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f \quad \longrightarrow \quad f_t(p) \approx \int_{\mathbb{M}_3} \exp\left(-\frac{d(p,q)^2}{2t}\right) \cdot f_0(q)\,\mathrm{d}q$$

$$\frac{\partial f}{\partial t} = -\frac{1}{2}\|\nabla f\|^2 \quad \longrightarrow \quad f_t(p) = \inf_{q \in \mathbb{M}_3} \frac{d(p,q)^2}{2t} + f_0(q)$$

- However, the Riemannian distance d, in general, is **expensive** to compute.
- In Portegies et al. [7] it is instead suggested to use the **cheap** *mav distance*.

Mav Distance

- Both the diffusion and dilation/erosion PDEs on $\mathbb{M}_3$ can be (approximately) solved using the *Riemannian distance* $d : \mathbb{M}_3 \times \mathbb{M}_3 \to \mathbb{R}_{\geq 0}$:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\Delta f \qquad \longrightarrow \quad f_t(p) \approx \int_{\mathbb{M}_3} \exp\left(-\frac{d(p,q)^2}{2t}\right) \cdot f_0(q)\,\mathrm{d}q$$

$$\frac{\partial f}{\partial t} = -\frac{1}{2}\|\nabla f\|^2 \quad \longrightarrow \quad f_t(p) = \inf_{q \in \mathbb{M}_3} \frac{d(p,q)^2}{2t} + f_0(q)$$

- However, the Riemannian distance d, in general, is **expensive** to compute.
- In Portegies et al. [7] it is instead suggested to use the **cheap** *mav distance*.
- We saw an application of the mav distance as a *learnable* invariant in PONITA, with the metric parameter $w_1, \ldots, w_4$ acting as trainable weights.

Mav Distance

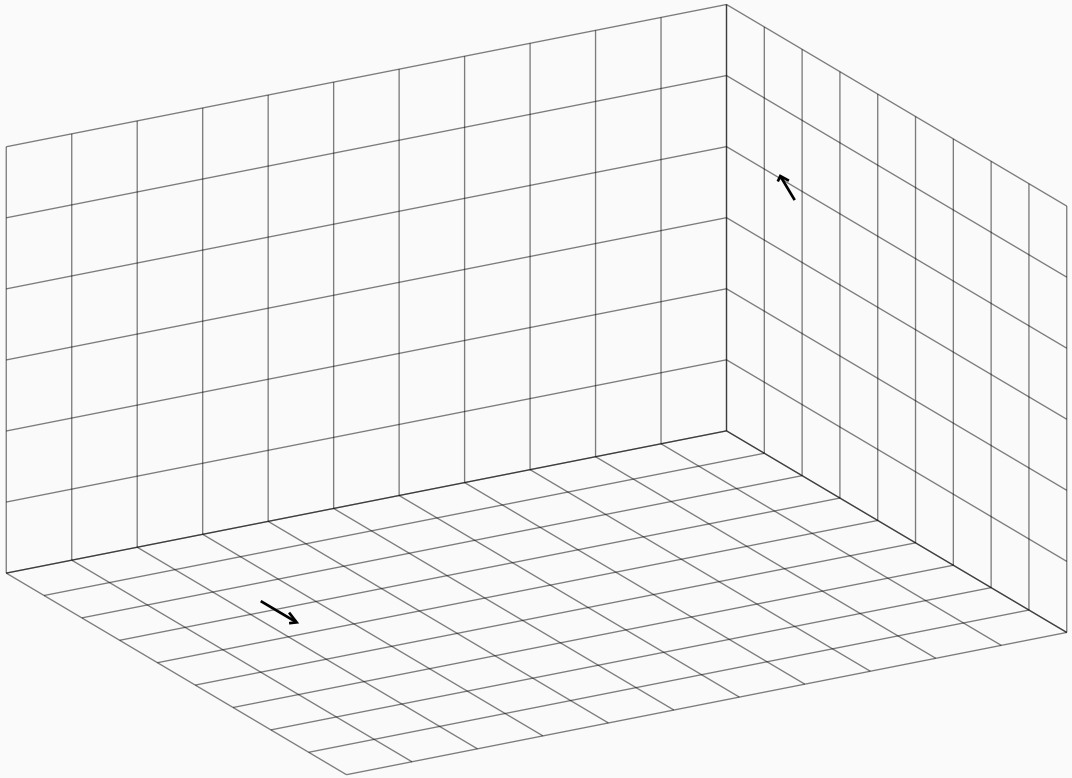
- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.

Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.

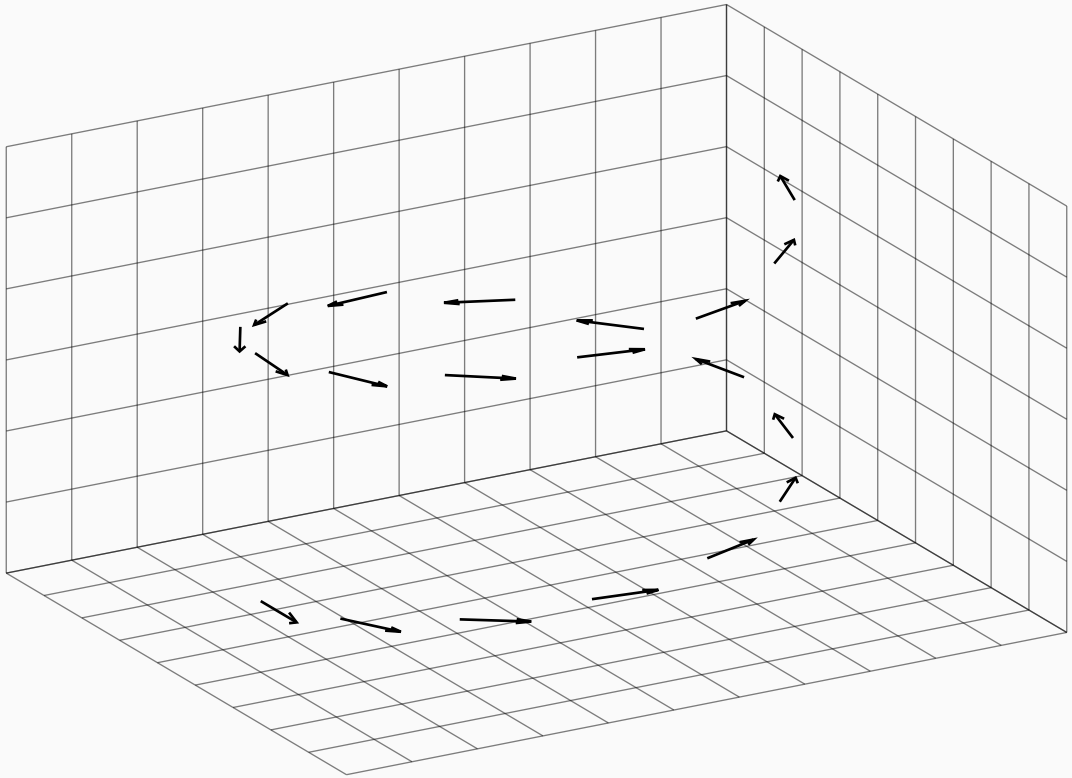
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



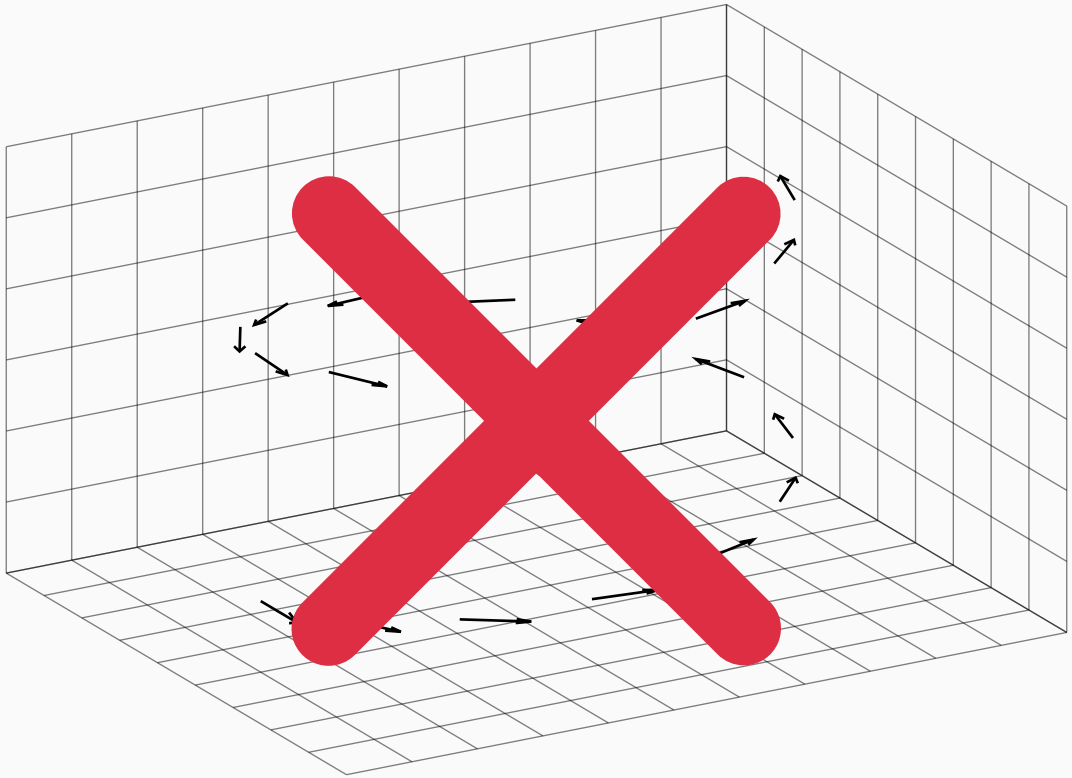
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



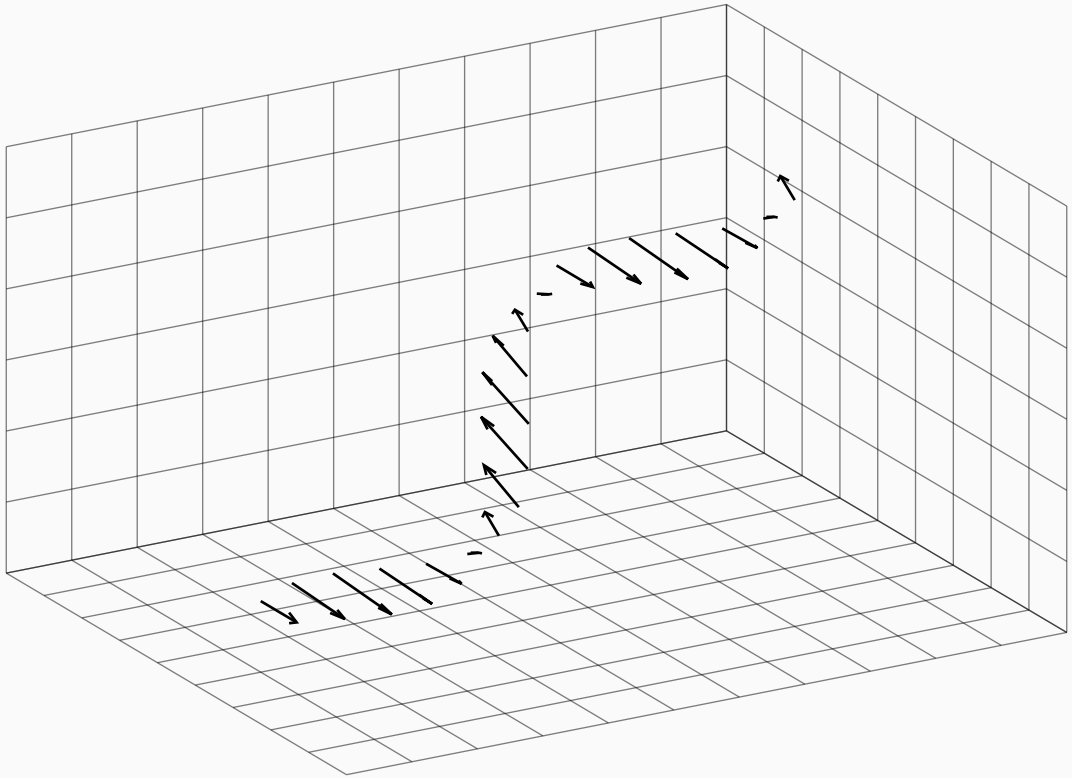
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



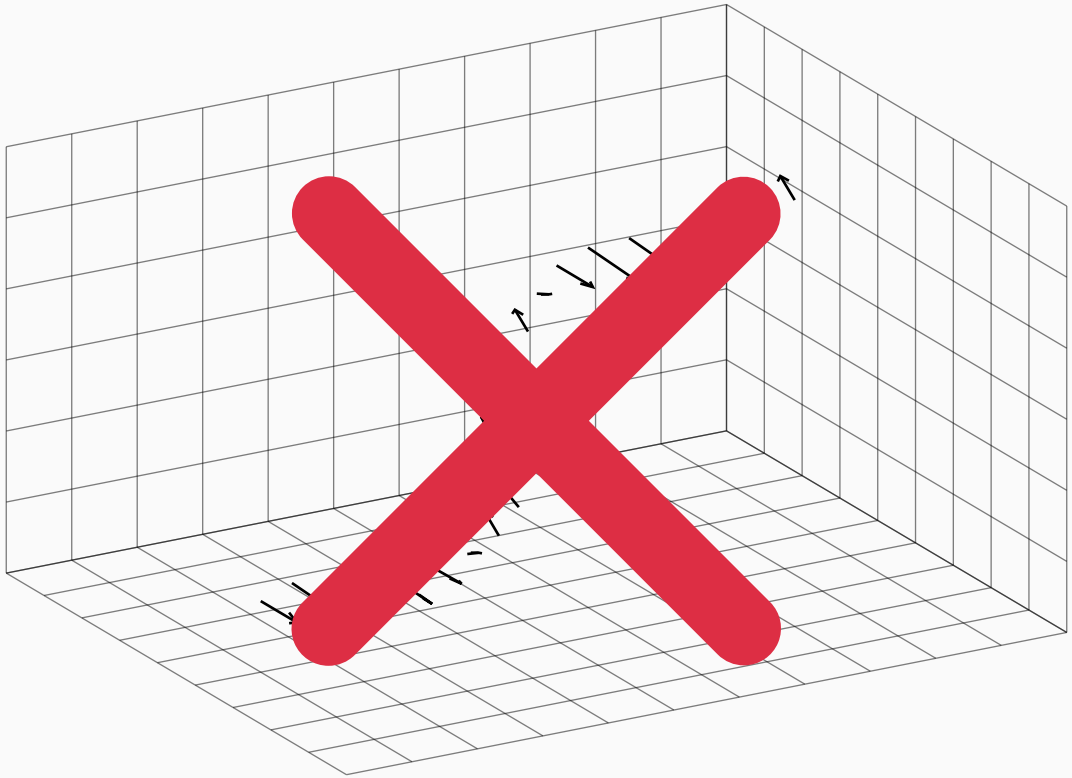
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



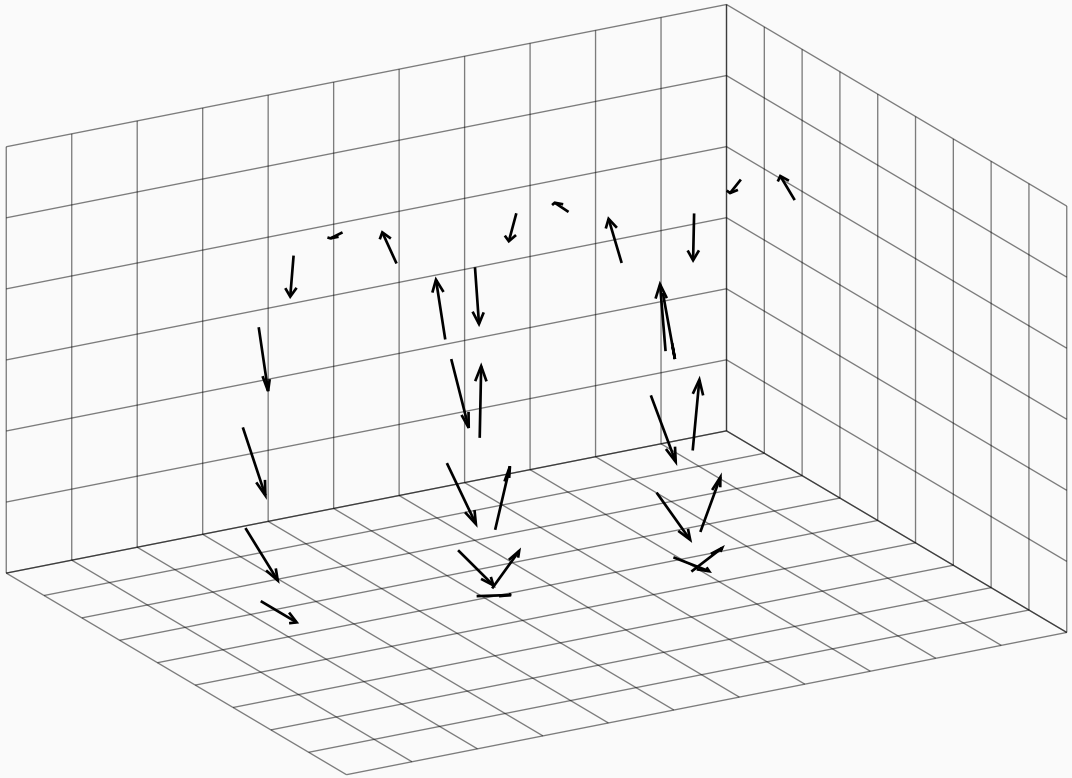
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



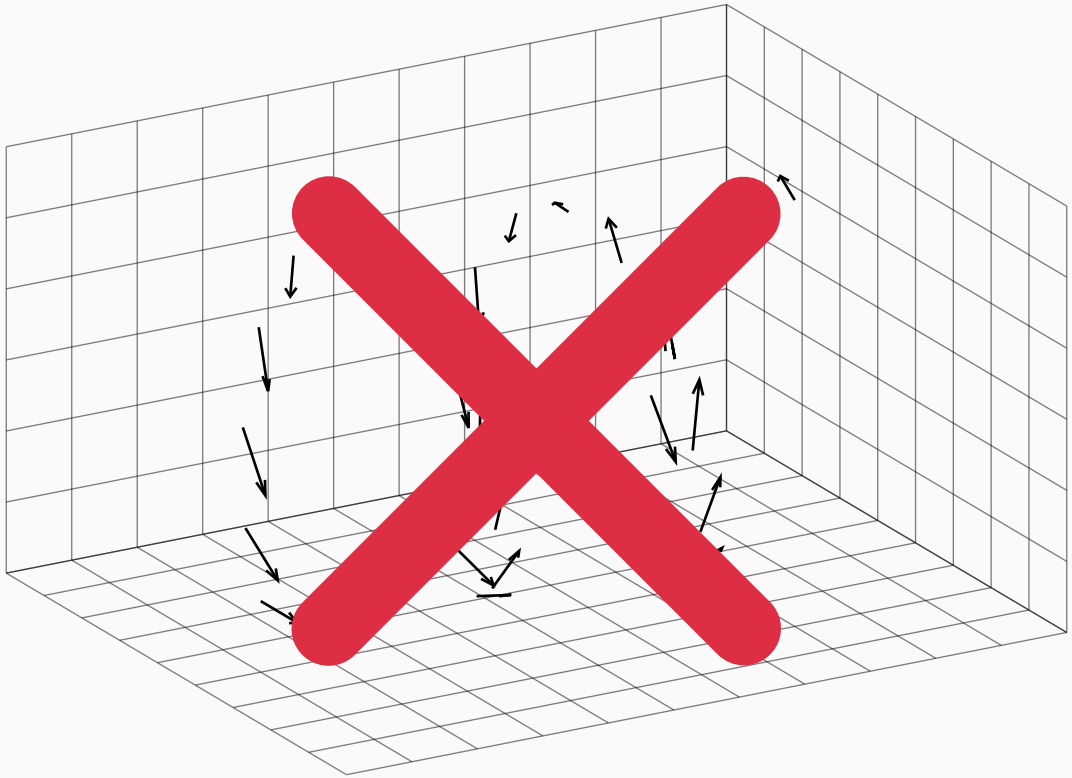
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



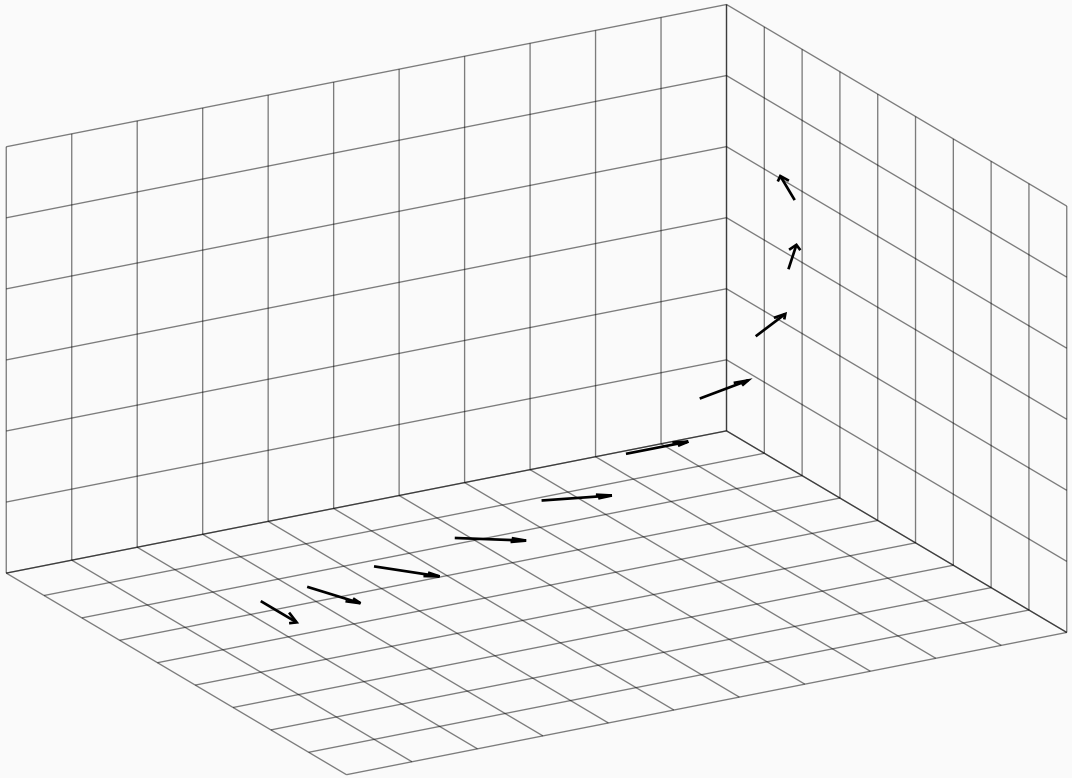
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



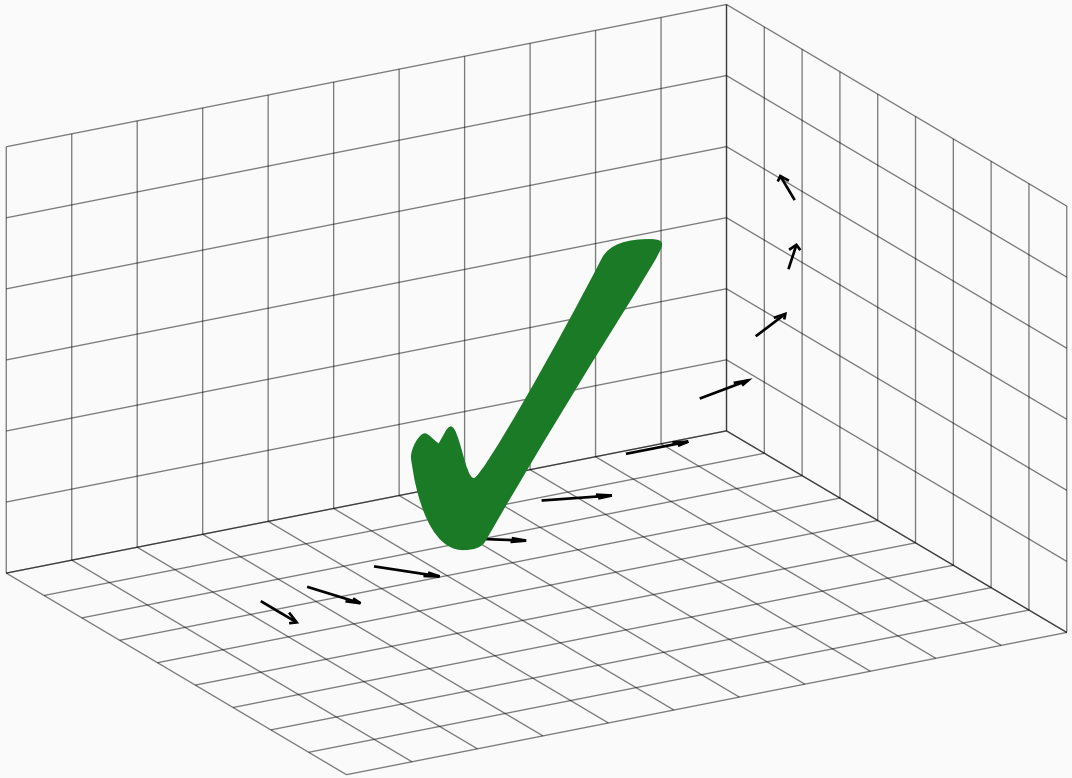
Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



Mav Distance

- The Riemannian distance is the length of the shortest curve γ between two points p_0, p_1 in $\mathbb{M}_3$.
- The mav distance is the length of the curve $p_t = \exp(tM) \triangleright p_0$ where $M \in \mathfrak{e}(3)$ is the **minimal angular velocity generator** between p_0 and $\exp(M) \triangleright p_0 = p_1$.



Mav Distance

Target	Unit	Original	Mav Distance (ours)	Difference %
μ	D	0.0195	0.0181	−07.2
α	a_0^3	0.0556	0.0540	−02.9
$\varepsilon_{\text{homo}}$	eV	0.0225	0.0229	+01.8
$\varepsilon_{\text{lumo}}$	eV	0.0205	0.0207	+01.0
$\Delta\varepsilon$	eV	0.0414	0.0431	+04.0
$\langle R^2 \rangle$	a_0^2	0.4160	0.4942	+18.8
ZPVE	meV	1.5647	1.5613	−00.2
U_0	eV	0.9920	0.7047	−28.9
U	eV	1.3593	1.0947	−19.5
H	eV	1.0204	1.0856	+06.4
G	eV	1.1856	0.9691	−18.3
c_v	cal/mol·K	0.0291	0.0283	−02.8

PONITA trained to predict chemical properties of various molecules (QM9 dataset [5], [6]). Mean absolute error on the test set is reported (lower is better).

Using the **mav distance** has a **marginal positive** impact on the accuracy of the PONITA model when predicting molecular properties.

Using the **mav distance** has a **marginal positive** impact on the accuracy of the PONITA model when predicting molecular properties.[1]

[1]However, it did have a significant negative impact on the financial situation of our group.

Thank you for your attention!

Questions?

Bibliography

[1] E. J. Bekkers, S. Vadgama, R. Hesselink, P. A. van der Linden, and D. W. Romero, "Fast, Expressive SE(3) Equivariant Networks through Weight-Sharing in Position-Orientation Space," in *ICLR*, 2024. [Online]. Available: https://openreview.net/forum?id=dPHLbUqGbr

[2] N. Thomas *et al.*, "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds," *arXiv preprint*, 2018, [Online]. Available: https://arxiv.org/abs/1802.08219

[3] B. Anderson, T. S. Hy, and R. Kondor, "Cormorant: Covariant Molecular Neural Networks," in *NeurIPS*, 2019, p. . [Online]. Available: https://papers.neurips.cc/paper_files/paper/2019/hash/03573b32b2746e6e8ca98b9123f2249b-Abstract.html

[4] G. Bellaard, B. M. N. Smets, and R. Duits, "Universal Collection of Euclidean Invariants between Pairs of Position-Orientations." [Online]. Available: https://arxiv.org/abs/2504.03299

[5] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific Data*, vol. 1, no. 1, 2014, doi: 10.1038/sdata.2014.22.

Bibliography

[6] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17," *Journal of Chemical Information and Modeling*, vol. 52, no. 11, pp. 2864–2875, Nov. 2012, doi: 10.1021/ci300415d.

[7] J. Portegies, G. Sanguinetti, S. Meesters, and R. Duits, "New Approximation of a Scale Space Kernel on SE(3) and Applications in Neuroimaging," in *SSVM*, 2015, pp. 40–52. doi: 10.1007/978-3-319-18461-6_4.

[8] R. Duits and E. Franken, "Left-Invariant Diffusions on the Space of Positions and Orientations," *IJCV*, vol. 92, no. 3, pp. 231–264, May 2011, doi: 10.1007/s11263-010-0332-z.

[9] B. M. N. Smets, J. Portegies, E. St-Onge, and R. Duits, "Total Variation and Mean Curvature PDEs on the Homogeneous Space of Positions and Orientations," *JMIV*, vol. 63, no. 2, pp. 237–262, 2021, doi: 10.1007/s10851-020-00991-4.

[10] G. Bellaard and B. M. N. Smets, "Roto-Translation Invariant Metrics on Position-Orientation Space." [Online]. Available: https://arxiv.org/abs/2504.03309

Some PONITA Details

- QM9 has graphs $(\mathcal{V}_0 \subset \mathbb{R}^3, \mathcal{E}_0)$ with some scalar features $(f_x)_{x\in\mathcal{V}_0} \subset \mathbb{R}$.
- Discretize $S^2_n \subset S^2$ and let $\mathcal{V} := \mathcal{V}_0 \times S^2_n \subset \mathbb{M}_3$ be the vertices of a fully connected (lifted) graph.
- Lift features as $\tilde{f}_v = \tilde{f}_{(x,n)} = f_x$ for all $v \in \mathcal{V}$.
- A feature $(f_v)_{v\in\mathcal{V}} \subset \mathbb{R}$ can be represented in the continuous setting as

$$f := \sum_{v\in\mathcal{V}} f_v \delta_v.$$

- Consequently the linear operator works as

$$(\Phi f)(v) = \int_{\mathbb{M}_3} k_\theta(v,w) f(w)\,\mathrm{d}w = \sum_{w\in\mathcal{V}} k_\theta(v,w) f(w) \qquad \forall v \in \mathcal{V},$$

  which is a particular choice of *message passing* for a Graph Neural Network.
- For performance reasons reduce the graph based on spatial distance, i.e.

$$(\Phi f)(v) = \sum_{w\in\mathcal{N}(v)} k_\theta(v,w) f(w),$$

  with $\mathcal{N}(x,n) := \{(x',n') \in \mathcal{V} \mid \|x - x'\| < C\}$.