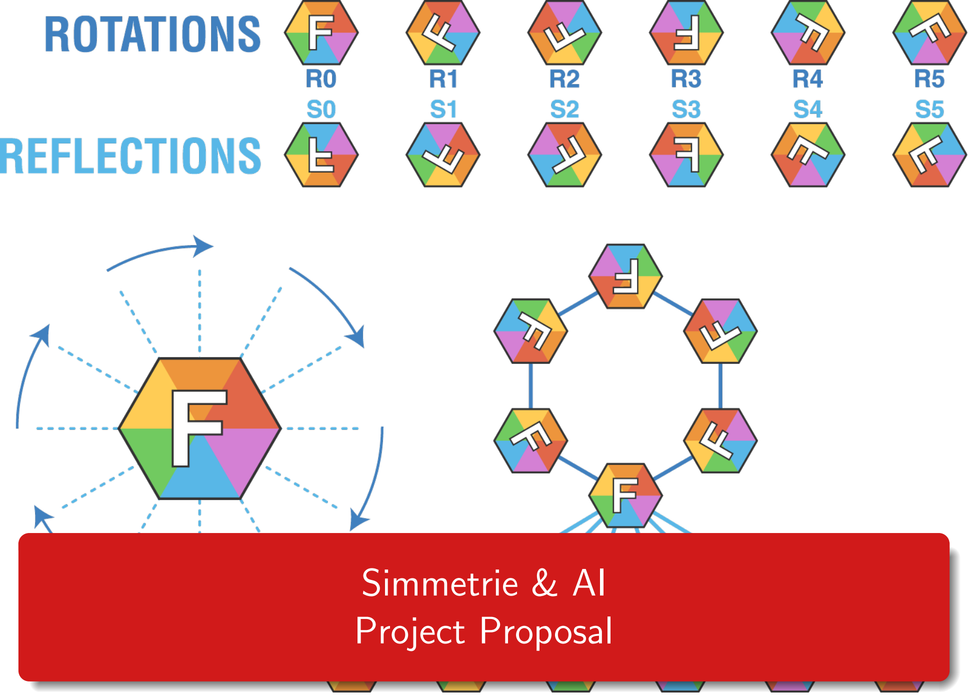


Simmetrie & AI
Project Proposal

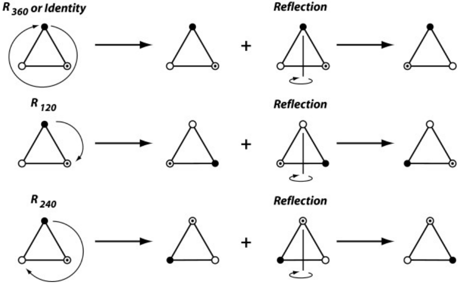
Outline

Fondamenti teorici

Simmetrie - Definizione intuitiva [9]

Definizione (intuitiva)

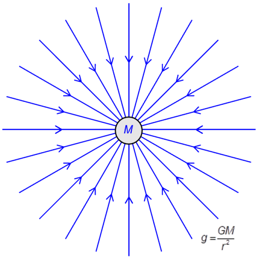
Un insieme di trasformazioni che lascia invariata una figura geometrica



Simmetrie - Fisica [9]

Legge di gravitazione di Newton

Dalle figure geometriche si può passare a studiare le simmetrie delle leggi fondamentali in fisica: ad esempio nella teoria della gravità di Newton un oggetto non cade verso il basso, ma verso il centro della terra. Questo perchè la legge di gravitazione di Newton non ha direzione privilegiata ma è invariante per rotazioni.



Simmetrie - Composizione e Gruppi [9]

Gruppo

Si consideri un set di trasformazioni $T_1, T_2, ...$ che lascia le leggi della fisica invarianti. Si applichi la trasformazioni T_j e poi la trasformazione T_i. La trasformazione composta sarà $T_i \cdot T_i$

Regole

Proprietà associativa $(g_\alpha \cdot g_\alpha) \cdot g_\gamma = g_\alpha \cdot (g_\alpha \cdot g_\gamma)$

Esistenza dell'elemento identità $I \cdot g_\alpha = g_\alpha$ e $g_\alpha \cdot I = g_\alpha$

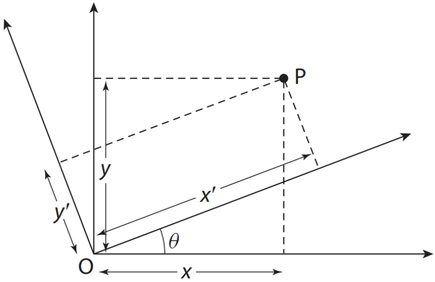
Esistenza dell'inversa $g_\alpha^{-1} \cdot g_\alpha = I$ e $g_\alpha \cdot g_\alpha^{-1} = I$

Remark

Non è richiesta la proprietà commutativa

$$\begin{cases} x' = \cos\theta x + \sin\theta y \\ y' = -\sin\theta x + \cos\theta y \end{cases}$$

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$



$R(\theta) \approx I + A$ dove A è un matrice infinitesimale di ordine θ

Quindi

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$$R = I + \theta J + \mathcal{O}(\theta^2)$$

$$R(\theta) = \lim_{N \to \infty} \left(R\left(\frac{\theta}{N}\right)\right)^N = \lim_{N \to \infty} \left(1 + \frac{\theta J}{N}\right)^N = e^{\theta J}$$

Simmetrie - Rotazioni - Approccio di Lie [9]

Si dimostra che

$$e^{\theta J} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Nel caso precedente l'invariante erano le distanze

$$dx^2 + dy^2 = dx'^2 + dy'^2$$

Se si generalizza questo vincolo a più dimensioni si ottengono le matrici di rotazione in più dimensioni

Si consideri la seguente espressione

$$RR'R^{-1} \approx (I + A)R(I - A) \approx R' + AR' - R'A$$

$$R' + B$$

$$RR'R \approx I + B + AB - BA$$

$$[A, B] = AB - BA$$

L'importanza del commutatore si apprezza considerando

$$A = i \sum_i \theta_i J_i \quad B = i \sum_i \theta_i' J_i$$

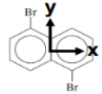
$$[J_i, J_j] = ic_{ijk} J_k$$

Con l'ultima relazione posso generare le rotazioni per n dimensioni

Transformation Matrices

Each symmetry operation can be represented by a 3×3 matrix that shows how the operation transforms a set of *x*, *y*, and *z* coordinates

Let's consider C_{2h} $\{E, C_2, i, \sigma_h\}$:



1,5-dibromonaphthalene

C_2

$$\begin{array}{l} x' = -x \\ y' = -y \\ z' = z \end{array} \quad \underset{\text{transformation matrix}}{\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -x \\ -y \\ z \end{bmatrix}$$

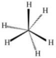
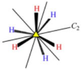
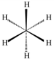
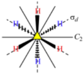
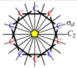
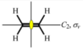
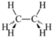
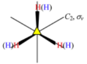
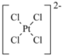
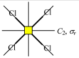
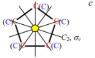
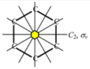
$$\begin{bmatrix} \text{new} \\ \text{coordinates} \end{bmatrix} = \begin{bmatrix} \text{transformation} \\ \text{matrix} \end{bmatrix} \begin{bmatrix} \text{old} \\ \text{coordinates} \end{bmatrix} = \begin{bmatrix} \text{new in terms} \\ \text{of old} \end{bmatrix}$$

i

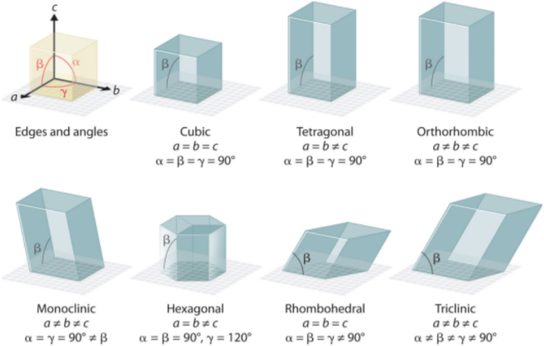
$$\begin{array}{l} x' = -x \\ y' = -y \\ z' = -z \end{array} \quad \underset{\text{transformation matrix}}{\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -x \\ -y \\ -z \end{bmatrix}$$

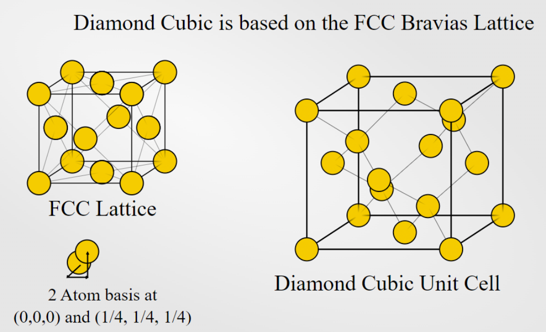
Simmetrie - Fisica/Chimica [9, 5]

Molecule	Projection[a,b]	Symmetry Elements	Point Group
 ethane(partial staggered)		C_3, $3C_2$	D_3
 ethane(staggered)		$C_3(S_6)$, $3C_2$, $3\sigma_d$, i	D_{3d}
		$C_5(S_{10})$, $5C_2$,$5\sigma_d$, i	D_{5d}
		$3C_2$, σ_h, $2\sigma_v$, i	D_{2h}
 ethane(eclipsed)		$C_3(S_3)$, $3C_2$, σ_h, $3\sigma_v$,	D_{3h}
		$C_4(S_4)$, $4C_2$, σ_h, $4\sigma_v$, i	D_{4h}
	 [c]	$C_5(S_5)$, $5C_2$, σ_h, $5\sigma_v$,	D_{5h}
		$C_6(S_6)$, $6C_2$, σ_h, $6\sigma_v$, i	D_{6h}

Crystal Systems [2]



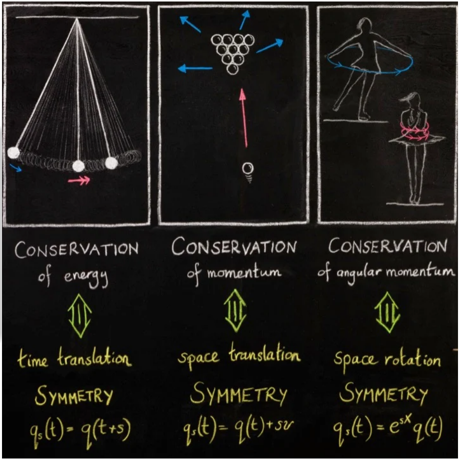
Crystal Systems [2]



Teorema di Noether [9, 3]

Teorema di Noether

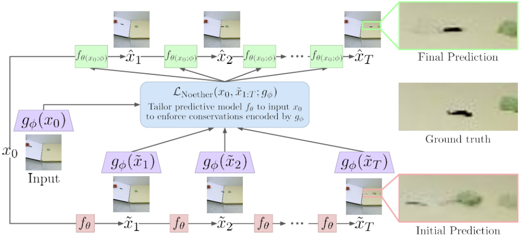
A ogni simmetria della lagrangiana, ovvero la funzione che definisce la dinamica (e quindi le variazioni di energia) di un sistema fisico, corrisponde una quantità conservata



Lavori precedenti

Noether networks: Meta-Learning useful Conserved Quantities [4]

$$\mathcal{L}_{Nother}\left(x_0, f_\theta\left(\tilde{x}_{1:T}\right)_{1:T}; g_\phi\right) = \sum_{t=1}^{T} \left|g_\phi(x_0) - g_\phi(\tilde{x}_t)\right|^2$$
$$\approx \sum_{t=1}^{T} \left|g_\phi(\tilde{x}_{t-1}) - g_\phi(\tilde{x}_t)\right|^2$$



Noether networks: Meta-Learning useful Conserved Quantities [4]

Algorithm 1 Prediction and training procedures for Noether Networks with a neural conservation loss

Given: predictive model class f; embedding model class g;
prediction horizon T; batch size N; training dist. $\mathcal{D}_{\text{train}}$;
learning rates λ_{in}, λ_{out}, λ_{emb};
task loss $\mathcal{L}_{\text{task}}$; Noether loss $\mathcal{L}_{\text{Noether}}$

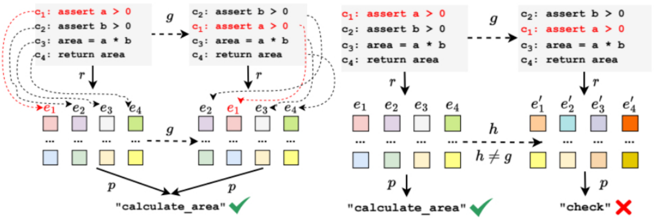
procedure PREDICTSEQUENCE($x_0; \theta, \phi$)
- $\tilde{x}_0, \hat{x}_0 \leftarrow x_0, x_0$
- $\tilde{x}_t \leftarrow f_\theta(\tilde{x}_{t-1})\ \forall t \in \{1, \ldots, T\}$ ▷ Initial predictions
- $\theta(x_0; \phi) \leftarrow \theta - \lambda_{\text{in}} \nabla_\theta \mathcal{L}_{\text{Noether}}(x_0, \tilde{x}_{1:T}; g_\phi)$ ▷ Inner step with Noether loss
- $\hat{x}_t \leftarrow f_{\theta(x_0;\phi)}(\hat{x}_{t-1})\ \forall t \in \{1, \ldots, T\}$ ▷ Final prediction with tailored weights
- **return** $\hat{x}_{1:T}$

procedure TRAIN
- $\phi \leftarrow$ randomly initialized weights ▷ Initialize weights for Noether embedding g
- $\theta \leftarrow$ randomly initialized weights ▷ Initialize weights for predictive model f
- **while** not done **do**
  - Sample batch $x_{0:T}^{(0)}, \ldots, x_{0:T}^{(N)} \sim \mathcal{D}_{\text{train}}$
  - **for** $0 \leq n \leq N$ **do**
    - $\hat{x}_{1:T}^{(n)} \leftarrow$ PREDICTSEQUENCE($x_0^{(n)}; \theta, \phi$)
  - $\phi \leftarrow \phi - \lambda_{\text{emb}} \nabla_\phi \sum_{n=0}^{N} \mathcal{L}_{\text{task}}(\hat{x}_{1:T}^{(n)}, x_{1:T}^{(n)})$ ▷ Outer step for embedding
  - $\theta \leftarrow \theta - \lambda_{\text{out}} \nabla_\theta \sum_{n=0}^{N} \mathcal{L}_{\text{task}}(\hat{x}_{1:T}^{(n)}, x_{1:T}^{(n)})$ ▷ Outer step for predictive model
- **return** ϕ, θ

Exploing code symmetries for learning program semantics [6]

Definizione Simmetria di un codice

Dato un blocco di codice c e un insieme di simmetrie G, un LLM deve assicurare che $\forall g \in G, m(g(c)) = m(c)$ e.g. x=2;y=4 and y=4;x=2



(a) SYMC (b) Existing LLMs

Definizione: Exploing code symmetries for learning program semantics [6]

Code space

L'insieme dei pezzi di codice

Definizione: Code representation unit

Un set di n istruzioni estratto da un set di istruzioni I, $i \in I^n$. Il *code space* I^n è l'insieme di tutte le *code representation unit* di interesse.

Exploing code symmetries for learning program semantics [6]

Definizione: Representation learning for code

Apprendimento di una funzione r che mappa un code representation unit $c \in I^n$ su un punto nel code representation space $\mathbb{R}^{d \times n}$, $r : I^n \longrightarrow \mathbb{R}^{d \times n}$ dove d è la dimensione del vettore in cui ciascuna istruzione viene mappata

Definizione: Predictive learning for code

Apprendimento di una funzione $p : \mathbb{R}^{d \times n} \longrightarrow \mathbb{R}^L$ che mappa la rappresentazione di un codice prodotta da dal *Representation learning* r nello spazio delle *label* $\mathbb{R}^L$

Exploing code symmetries for learning program semantics [6]

Examples

Neural network: i layer iniziali di una rete neurale servono come representation learning function, i successivi come predictive learning function p. Si ha quindi $p \circ r$

Exploing code symmetries for learning program semantics [6]

Definizione: G-equivariant code representation learning

Sia G un gruppo di simmetria che consiste in un gruppo di trasformazioni che preservano la semantica applicato a un code representation unit $c \in I^n$. Una funzione di rappresentazione r: $I^n \longrightarrow \mathbb{R}^{d \times n}$ è G-equivariant se $\forall g \in G$ e $\forall c \in I^n$, abbiamo $g \circ r(c) = r(g \circ c)$

Definizione: G-invariant code predictive learning

Sia G un gruppo di simmetria che consiste in un gruppo di trasformazioni che preservano la semantica applicato un vettore di rappresentazione del programma $c \in I^n$. Una predictive leraning function p: $\mathbb{R}^{d \times n} \longrightarrow \mathbb{R}^L$ è G-invariant se $\forall g \in G$ e $\forall e \in \mathbb{R}^{d \times n}$, abbiamo $p(g \circ e) = p(e)$

Exploing code symmetries for learning program semantics [6]

Definizione: program interpretation graph

Dato un code representation unit, possono esserci diversi modi di esecuzione ciascuno corrispondente alla struttura composizionale dell'interprete del programma $\{f_1, ..., f_n\}$. Si può quindi costruire il grafo di interpretazione $\mathcal{IG} = (V, E)$ considerando tutti i possibili metodi di esecuzione di c. Nel grafo i nodi V_i corrispondono a f_i, mentre segmenti E_{ij} orientati corrispondono un metodo di esecuzione

Definizione: G-invariant code predictive learning

Sia G un gruppo di simmetria che consiste in un gruppo di trasformazioni che preservano la semantica applicato un vettore di rappresentazione del programma $c \in I^n$. Una predictive learning function p: $\mathbb{R}^{d \times n} \longrightarrow \mathbb{R}^L$ è G-invariant se $\forall g \in G$ e $\forall e \in \mathbb{R}^{d \times n}$, abbiamo $p(g \circ e) = p(e)$

Exploing code symmetries for learning program semantics [6]

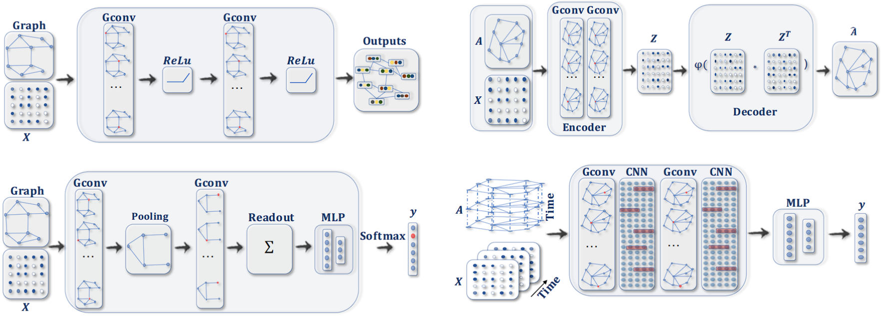
Remark

In questo modo si traduce il codice in un oggetto studiabile tramite la teoria dei gruppi di simmetria.

	Violation
SYMC	0%
GPT-4	43%
WizardCoder	14%
code2vec	61%
code2seq	52%
GGNN	7%

Graph neural networks

Graph neural networks [7, 8]



Lavori interessanti

Lavori interessanti

- KRIPPENDORF, Sven; SYVAERI, Marc. Detecting symmetries with neural networks. Machine Learning: Science and Technology, 2020, 2.1: 015010.
- FARINA, Francesco; SLADE, Emma. Symmetry-driven graph neural networks. arXiv preprint arXiv:2105.14058, 2021.

Bibliography I

[1] https://upload.wikimedia.org/wikipedia/commons/9/95/Tranformation_matrices.png.

[2] https://chem.libretexts.org/.

[3] https://www.iltascabile.com/scienze/emmy-noether-matematica.

[4] Ferran Alet et al. "Noether networks: meta-learning useful conserved quantities". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 16384–16397.

[5] Lan Chen, Hongwei Sun e Chengming Lai. "Teaching Molecular Symmetry of Dihedral Point Groups by Drawing Useful 2D Projections". In: *Journal of Chemical Education* 92.8 (2015), pp. 1422–1425.

Bibliography II

[6] Kexin Pei et al. "Exploiting Code Symmetries for Learning Program Semantics". In: (2023).

[7] Patrick Reiser et al. "Graph neural networks for materials science and chemistr". In: *Communications Materials* 3.1 (2022), p. 93.

[8] Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[9] A. Zee. *Group Theory in a Nutshell for Physicists*. In a Nutshell. Princeton University Press, 2016. ISBN: 9780691162690.