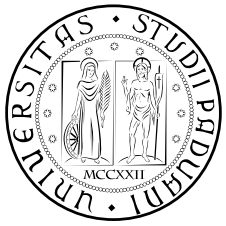


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCATRONICA

TESI DI LAUREA MAGISTRALE

ANALISI COMPARATIVA DELLE PIÙ RECENTI ARCHITETTURE DI RETI NEURALI PER LA PREVISIONE DEL MOVIMENTO UMANO IN ROBOTICA COLLABORATIVA

Relatore: Stefano Michieletto

Correlatore: Monica Reggiani

Correlatore: Michael Vanuzzo

Laureando: Davide Carollo
2020223-IMC

ANNO ACCADEMICO: 2022-23

ABSTRACT

La robotica collaborativa è un settore in continua evoluzione che si sta sviluppando molto velocemente negli ultimi anni. Mediante un approccio che sfrutti appieno le capacità uniche degli esseri umani e dei robot, sarà possibile creare un ambiente di lavoro flessibile che soddisfi le esigenze in continuo mutamento della produzione industriale. Un aspetto chiave per migliorare le capacità collaborative dei robot è la possibilità di prevedere con precisione i movimenti umani, permettendo così una collaborazione più fluida e una maggiore efficienza.

Sebbene il problema della previsione del movimento umano abbia ricevuto notevole attenzione negli ultimi anni, non esistono ancora delle analisi estensive nello specifico contesto della robotica collaborativa. Il principale obiettivo di questa tesi è quindi il confronto delle prestazioni di alcune delle architetture di reti neurali più recenti per la predizione del movimento umano.

Uno dei limiti principali di queste architetture è il fatto che sono sviluppate per predire il movimento in contesti generici. Per tale ragione, non esistono attualmente delle analisi che dimostrino quale strategia risulti più adatta in ambiti specifici e di interesse come la collaborazione tra uomo e robot. Esaminare questi aspetti è essenziale poiché, ottenere una previsione dell'immediato futuro, cioè con orizzonte temporale di alcuni secondi, risulta estremamente vantaggioso: sapere come si muoverà l'operatore, permette al robot di effettuare movimenti in modo più efficiente e sicuro, portando così ad una maggiore adozione di questa tecnologia.

Un altro obiettivo di questa tesi consiste nel superare una delle limitazioni principali per lo sviluppo dei modelli di *deep learning*, che rappresentano attualmente lo stato dell'arte per il problema della predizione del movimento umano, ovvero la mancanza di dataset completi e specifici per l'addestramento e la valutazione dei modelli nell'ambito della robotica collaborativa. È stato dunque raccolto un dataset focalizzato interamente sull'interazione tra essere umano e robot in un contesto industriale. Questo nuovo dataset è stato utilizzato per effettuare l'analisi e la comparazione delle architetture considerate, ricavando quali di queste portino i risultati migliori in diversi campi di applicazione secondo delle metriche di valutazione oggettive.

INDICE

1 INTRODUZIONE 1
- 1.1 Robotica collaborativa 1
- 1.2 Previsione del movimento umano 3
- 1.3 Stato dell'Arte 4
  - 1.3.1 Dataset di movimenti umani 4
  - 1.3.2 Rappresentazione della posa di una persona 5
  - 1.3.3 Reti neurali per la predizione del movimento umano 9

2 ARCHITETTURE PER LA PREVISIONE DEL MOVIMENTO 13
- 2.1 DMGNN 13
  - 2.1.1 Principio di funzionamento dell'Encoder 14
  - 2.1.2 Principio di funzionamento del Decoder 16
  - 2.1.3 Allenamento della rete 17
  - 2.1.4 Modifiche implementate 17
- 2.2 HRI 17
  - 2.2.1 Modello di Attenzione del Movimento 18
  - 2.2.2 Modello di Previsione 20
  - 2.2.3 Allenamento della rete 21
  - 2.2.4 Modifiche implementate 21
- 2.3 PVRED 23
  - 2.3.1 Principio di funzionamento di RED 23
  - 2.3.2 Principio di funzionamento di PVRED 24
  - 2.3.3 Trasformazione in Quaternioni 26
  - 2.3.4 Allenamento della rete 26
  - 2.3.5 Modifiche implementate 27

3 METRICHE 29
- 3.1 Descrizione delle Metriche 29
  - 3.1.1 Angoli di Eulero 29
  - 3.1.2 Differenza tra gli Angoli Articolari 30
  - 3.1.3 Errore di Posizione 30
  - 3.1.4 Valutazione Qualitativa 31
- 3.2 Modello base: Zero-Velocity 32
- 3.3 Metriche usate in letteratura 32

4 ESPERIMENTI 35
- 4.1 Analisi preliminare con H3.6M e AMASS 35
- 4.2 Setup sperimentale per la raccolta del dataset 37
- 4.3 Azioni registrate 41
- 4.4 Elaborazione dati 43

5 RISULTATI 47
- 5.1 Risultati su H3.6M 47
- 5.2 Risultati su AMASS 47
- 5.3 Risultati su CS_CORO 50

5.4 Sviluppi Futuri 52
Conclusioni 55

BIBLIOGRAFIA 59

ELENCO DELLE FIGURE

Figura 1 Posa di riferimento per il dataset AMASS (t-pose) e nome breve dei 24 giunti del modello SMPL. 6

Figura 2 Una posa del dataset H3.6M e nome breve dei 32 giunti del relativo modello. 6

Figura 3 Schema semplificato dell'architettura di DM-GNN, basata su un sistema Encoder-Decoder. L'Encoder è formato da una serie di blocchi chiamati *Multiscale Graph Computational Unit* (MG-CU) che elaborano le pose passate mentre il Decoder è composto da una *Graph-based Gate Recurrent Unit* (G-GRU) che prevede le pose future in modo ricorsivo. 14

Figura 4 Esempio numerico dell'estrazione di *key* (M pose), *value* (M + T pose) e *query* (M pose) dalla storia passata (N pose). 19

Figura 5 Schema del funzionamento di PVRED, tratto da [29]. Sia l'Encoder che il Decoder hanno in ingresso le pose x_n, le velocità v_n e le posizioni p_n. QT indica il layer dove avviene la trasformazione da asse-angolo a quaternioni. 24

Figura 6 Sensore inerziale MTw Awinda di XSens per la raccolta dei dati sulla posa della persona. Tra le informazioni raccolte sono compresi gli angoli articolari, le posizioni assolute di diversi punti del corpo e il centro di massa. 39

Figura 7 Guanti XSens Metagloves by Manus per la registrazione accurata delle posizioni e dei movimenti delle dita. 39

Figura 8 Una delle telecamere Azure Kinect utilizzate per registrare le azioni nel dataset. Ognuna dispone di un sensore di profondità, una videocamera a colori, un accelerometro, un giroscopio e un array di microfoni. 39

Figura 9 Foto dell'area operativa allestita per la raccolta del dataset. 40

Figura 10 Robot *Franka Emika Panda* utilizzato per la collaborazione con l'operatore nella raccolta del dataset. 40

Figura 11 Sequenza di azioni effettuate da uno dei soggetti registrati durante l'esecuzione dell'attività A. 43
Figura 12 Esempio del ricampionamento dei dati sui primi 100 frame di una registrazione. I timestamp sono in nanosecondi e vengono traslati in modo che il primo frame abbia timestamp = 0. Si può osservare che i timestamp originali non sono ad intervalli costanti mentre nel segnale ricampionato lo sono. 44
Figura 13 Risultati di DMGNN, HRI, PVRED e Zero-Velocity su H3.6M. 48
Figura 14 Risultati di DMGNN, HRI, PVRED e Zero-Velocity su AMASS. 49
Figura 15 Risultati di DMGNN, HRI, PVRED e Zero-Velocity su CS_CORO. 51

ELENCO DELLE TABELLE

Tabella 1 Metriche utilizzate dalle diverse architetture per la previsione del movimento umano. 33
Tabella 2 Elenco dei dataset di AMASS usati per l'analisi. I dati sono ottenuti considerando Mocap ricampionati ad un framerate di 60fps. 36
Tabella 3 Specifiche del Cluster di calcolo utilizzato per l'elaborazione dei dati. 37

1 INTRODUZIONE

1.1 ROBOTICA COLLABORATIVA

La robotica collaborativa rappresenta una delle più significative innovazioni nel campo dell'automazione industriale e dei sistemi robotici. I due attori principali in questo innovativo settore industriale, gli esseri umani e i robot, vengono infatti trattati in modo opposto rispetto alla concezione tradizionale di automazione. Mentre l'automazione "classica", ovvero quella non collaborativa, punta principalmente ad ottenere la massima efficienza nell'utilizzo dei robot, la robotica collaborativa tenta di riportare al centro dei processi produttivi l'essere umano, focalizzandosi su un approccio basato sulla flessibilità senza per questo diminuire la sicurezza o sacrificare drasticamente l'efficienza.

Questo innovativo campo di ricerca ha la potenzialità di rivoluzionare i processi industriali, la logistica, la sanità e molte altre aree, trasformando radicalmente la dinamica attualmente esistente tra uomini e macchine. Diventa infatti possibile l'integrazione di robot in ambienti di lavoro tradizionalmente riservati agli esseri umani, aprendo così nuove prospettive in quegli ambiti che richiedono precisione, ripetitività o lavori fisicamente impegnativi.

L'obiettivo principale è quello di sfruttare le capacità uniche degli esseri umani, come la creatività, la comprensione del contesto e la flessibilità decisionale, combinandole con la precisione e la forza dei robot. Un aspetto fondamentale da evidenziare è che i robot collaborativi sono progettati per operare in sicurezza a fianco degli esseri umani, sono infatti dotati di sensori e protocolli per prevenire incidenti e garantire un ambiente di lavoro sicuro.

I robot collaborativi, anche conosciuti come cobot, possono essere impiegati con successo in scenari quali:

- *Assemblaggio:* umani e cobot possono lavorare insieme nelle linee di assembleggio, dove il cobot può gestire compiti pesanti o ripetitivi, come il recupero delle parti, il posizionamento e le fasi di assemblaggio di base, mentre l'essere umano può eseguire fasi più intricate o complesse.

- *Controllo qualità:* in questo tipo di situazione il cobot può assistere nell'ispezione dei prodotti, nell'esecuzione di misurazioni o nell'identificazione dei difetti, mentre l'essere umano supervisiona il processo, prende decisioni e gestisce valutazioni più soggettive.

- *Movimentazione di materiale:* date le sue caratteristiche, il cobot può sollevare e trasportare oggetti pesanti, nel frattempo l'essere umano può concentrarsi su compiti di supervisionamento, coordinamento o interazione con altri lavoratori.
- *Imballaggio e pallettizzazione:* gestire attività ripetitive come posizionare gli articoli in scatole o impilarli su pallet è un altro scenario in cui l'utilizzo di un cobot può risultare vantaggioso, l'operatore può infatti gestire la movimentazione di prodotti dal formato non standardizzato, oltre ad occuparsi di monitorare il processo, garantire la qualità e gestire la logistica.
- *Asservimento macchine:* il cobot può caricare e scaricare materiali nelle macchine, monitorare i processi o eseguire attività di manutenzione ordinaria, mentre l'addetto supervisiona l'operazione e gestisce gli aspetti più complessi del funzionamento della macchina.

Questi sono solo alcuni esempi di come gli esseri umani possono collaborare con i cobot in ambienti industriali. Le potenziali applicazioni sono vaste e continuano ad espandersi man mano che la tecnologia avanza e le capacità dei robot collaborativi si evolvono. Tuttavia, oltre alle molte potenzialità che posseggono i cobot, bisogna anche considerarne alcune criticità, tra cui:

- *Formazione degli operatori:* gli operatori umani devono essere formati per lavorare in modo efficiente con i cobot. Questa formazione può richiedere tempo e risorse considerevoli, gli operatori devono infatti essere in grado di intervenire sul cobot per risolvere la maggior parte delle problematiche che possono presentarsi.
- *Efficienza:* in alcuni casi, la cooperazione uomo-robot può essere più lenta rispetto all'utilizzo di un normale lavoratore umano e di un robot in spazi di lavoro isolati. Ciò potrebbe essere dovuto a diversi motivi, come la complessità del compito, la necessità di coordinamento tra l'operatore e il cobot, o le limitazioni intrinseche della tecnologia attuale.
- *Normativa in vigore:* la stringente regolamentazione relativa alla coesistenza di esseri umani e robot nella stessa area di lavoro può avere un impatto negativo sulla cooperazione uomo-robot. Queste norme, in vigore per garantire la sicurezza dei lavoratori umani e prevenire incidenti, possono imporre vincoli al movimento e alle capacità dei robot, influenzandone così le prestazioni.

Per migliorare il comportamento del cobot in diverse situazioni, è importante migliorare la capacità di interpretare le intenzioni dell'essere umano. Una valida strategia consiste nel tentare di prevedere

il movimento dell'operatore al fine di migliorare la collaborazione, ottimizare le operazioni o evitare possibili situazioni di pericolo.

1.2 PREVISIONE DEL MOVIMENTO UMANO

L'anticipazione del movimento umano consiste nell'interpretare gli schemi secondo cui si muove una persona e tradurli in modelli predittivi in grado di calcolarne i movimenti futuri. Tramite una stima dell'incertezza è inoltre possibile prevedere uno spettro di possibili eventi futuri. Prevedere il movimento umano può portare benefici in molteplici applicazioni: nei veicoli a guida autonoma consente di predire le intenzioni dei pedoni in tempo reale per evitare incidenti, nello sport e nell'intrattenimento la previsione del movimento migliora le esperienze di realtà virtuale, le interazioni con i videogiochi e le tecnologie di motion capture. Vi sono inoltre applicazioni nel settore sanitario, che consentono di assistere nell'analisi e nella riabilitazione dei disturbi legati al movimento.

Purtroppo, prevedere con precisione il movimento del corpo rappresenta una sfida estremamente complessa, in ragione della natura non deterministica del comportamento umano. Infatti, i nostri movimenti sono costantemente influenzati da stimoli interni ed esterni, portando a cambiamenti del movimento improvvisi e difficilmente prevedibili. Ad esempio, se si osserva una persona camminare per alcuni secondi, la conclusione più probabile è prevedere che la camminata continui anche per i secondi successivi. Eventi improvvisi, come lo squillo del cellulare o il passaggio di un conoscente, possono modificare in modo significativo il movimento della persona, rendendo la previsione completamente sbagliata.

La capacità di comprendere e anticipare il movimento umano è un aspetto critico per i sistemi intelligenti che coesistono e collaborano con gli esseri umani in spazi di lavoro condivisi. Nello specifico campo della robotica collaborativa, la previsione del movimento umano è un aspetto fondamentale per garantire produttività e sicurezza durante le attività di collaborazione uomo-macchina. Tuttavia, prevedere il movimento umano in questo contesto è estremamente impegnativo a causa di diversi fattori che non sono presenti negli scenari finora considerati in letteratura. Bisogna infatti considerare che nello spazio di lavoro condiviso si trovano sia oggetti statici che in movimento, inoltre, va tenuto conto della presenza del cobot stesso. Questi elementi influenzano in modo significativo il comportamento umano e aggiungono un ulteriore livello di complessità al problema.

Gli algoritmi di *deep learning* più recenti hanno portato ad ottimi risultati nella previsione del movimento, superando i metodi statistici e gli algoritmi di *machine learning* tradizionali. Tuttavia, la maggior parte dei dataset esistenti, e di conseguenza i modelli addestrati su di essi, si basano su registrazioni di esseri umani che svolgono attività

generiche in spazi aperti, privi di oggetti ed ostacoli.

Uno degli obiettivi di questa tesi è quindi la raccolta di un dataset incentrato sui movimenti di un operatore in una stazione di assemblaggio e sulla sua collaborazione con un robot, cioè in un contesto industriale. Tale dataset verrà poi utilizzato per effettuare l'analisi di alcune delle reti neurali più recenti nell'ambito della previsione del movimento umano. L'obiettivo consiste nel ricavare quale tra queste architetture fornisca i risultati migliori secondo diversi parametri come ad esempio la precisione della posizione delle articolazioni più estreme del corpo, come le mani, o la bontà complessiva della postura predetta.

1.3 STATO DELL'ARTE

1.3.1 *Dataset di movimenti umani*

Sono attualmente presenti diversi dataset riguardanti i movimenti corporei, ognuno con le proprie peculiarità e applicazioni distintive. Di seguito sono presentati i dataset più significativi, evidenziando le loro caratteristiche principali, i vantaggi e le possibili limitazioni.

CMU

CMU Motion Capture [4] è un dataset che fornisce le pose 3D di 144 soggetti diversi per un totale di 2605 registrazioni. Contiene un ampio spettro di movimenti, inclusi movimenti quotidiani come camminare e correre, nonché movimenti sportivi come arrampicarsi e ballare. Nell'ambito della predizione del movimento umano, vengono spesso scartate diverse azioni, per esempio quelle di interazioni tra più persone e quelle che non forniscono dati sufficienti per allenare delle reti neurali. Ciò consente anche di essere consistenti con altri lavori basati su questo dataset, ad esempio [9].

3DPW

3D Poses in the Wild Dataset [28] è il primo dataset raccolto all'aperto con pose 3D accurate, registrate dalla fotocamera di un telefono in movimento. Questo nuovo approccio consente una cattura più dinamica del movimento umano, soprattutto in ambienti esterni. Con 60 sequenze video e 18 modelli 3D, include una vasta gamma di abbigliamenti e scene complesse. La complessità di questo dataset è al tempo stesso un punto di forza e una debolezza: presenza di telecamere in movimento, deriva dei sensori inerziali, occlusioni e più persone visibili nel video sono dei fattori che distinguono questo dataset dagli altri, tuttavia, potrebbero richiedere metodi di elaborazione e analisi più avanzati.

H3.6M

Human 3.6 Million [8, 5] è uno dei primi e più consolidati dataset nel campo, contenente 3,6 milioni di frame di pose umane 3D e le corrispondenti immagini. I dati presentano 11 attori professionisti che agiscono in 15 diversi scenari. Si tratta di un set di dati ricco, altamente diversificato e che offre un buon equilibrio tra attori maschili e femminili. Le ridotte dimensioni del dataset possono risultare uno svantaggio, poiché potrebbero non essere sufficienti per i modelli di deep learning più avanzati, al contrario di dataset più vasti come ad esempio AMASS. Inoltre, attualmente non è più accessibile tramite fonti ufficiali. Infatti, la maggior parte delle ricerche più recenti fa uso di una versione rivista di questo dataset, la quale comprende solamente un sottoinsieme di 7 soggetti sugli 11 totali, per un numero complessivo di approssimativamente un milione di pose. Nonostante ciò, Human3.6M è ancora utilizzato frequentemente grazie alle sue posizioni articolari 3D altamente precise, ai diversi scenari e alle grandi variazioni tra le pose.

AMASS

Archive of Motion Capture As Surface Shapes [14] è una raccolta ampia e diversificata di dati sul movimento umano, che aggrega 15 diversi dataset di motion capture basati su marcatori ottici in una struttura e parametrizzazione comuni. Il vantaggio principale è la sua dimensione, con oltre 40 ore di dati di movimento, oltre 300 soggetti e più di 11.000 movimenti. Per fare un paragone, le dimensioni di AMASS sono approssimativamente 14 volte quelle di H3.6M. AMASS rappresenta quindi un'importante risorsa per le applicazioni di deep learning. Il dataset è standardizzato utilizzando il modello *Skinned Multi-Person Linear Model* (SMPL) [11], garantendo coerenza e compatibilità tra i vari movimenti. Un ulteriore aspetto significativo di AMASS è la sua natura in continua evoluzione: è possibile integrare nuovi dataset, aumentando in questo modo la quantità di dati disponibili e, di conseguenza, le potenzialità dei modelli addestrati.

1.3.2 *Rappresentazione della posa di una persona*

La posizione in cui si trova il corpo di un essere umano può essere descritta tramite due diverse modalità:

- *Posizioni assolute dei giunti:* dato un sistema di riferimento cartesiano globale, ogni punto di interesse del corpo viene descritto tramite la sua posizione assoluta, espressa dalla terna di coordinate (x, y, z).

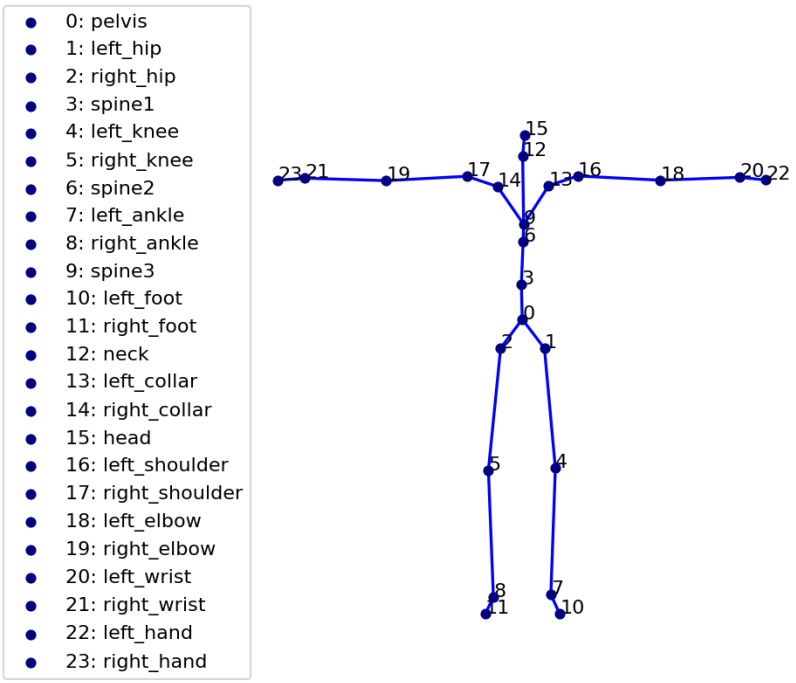


Figura 1: Posa di riferimento per il dataset AMASS (t-pose) e nome breve dei 24 giunti del modello SMPL.

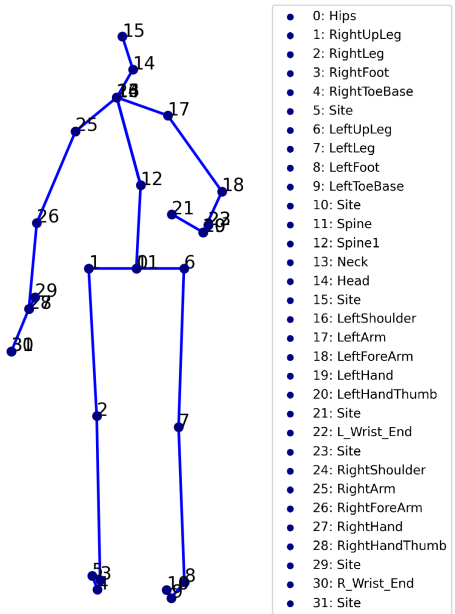


Figura 2: Una posa del dataset H3.6M e nome breve dei 32 giunti del relativo modello.

- *Angoli dei giunti:* data una posa di partenza, tutte le altre pose vengono descritte tramite l'angolo che ogni articolazione del corpo deve fare per allineare la posa base con quella in esame.

In Figura 1 e in Figura 2 si possono vedere rispettivamente la posa base utilizzata nel dataset AMASS e una posa del dataset H3.6M; viene inoltre indicato il nome, semplificato, delle articolazioni considerate. Le catene dei giunti sopra illustrate differiscono non solo per il numero di articolazioni complessive, ma anche per altri fattori: degli esempi sono l'orientazione di alcuni giunti (si confronti la zona del bacino le cui posizioni sono le stesse a prescindere dalla posa della persona) e la lunghezza di diverse componenti dello scheletro (si veda la posizione dei giunti della schiena).

Sebbene la rappresentazione tramite posizioni assolute sia più semplice e diretta, in ambito di previsione del movimento si preferisce utilizzare la descrizione per mezzo degli angoli per i seguenti motivi:

- Consente l'utilizzo diretto dei risultati ottenuti da persone di taglia diversa, per le quali lo stesso movimento porta agli stessi angoli ma a posizioni assolute differenti.
- Garantisce che i risultati delle previsioni effettuate dalle reti neurali portino ad una posa in cui lo scheletro mantiene tutte le lunghezze dei segmenti che lo compongono.
- Permette di passare correttamente alla rappresentazione tramite posizioni assolute senza ambiguità sulla posizione dei giunti (tramite un calcolo di cinematica diretta). L'inverso (cinematica inversa) non è invece possibile senza perdita di informazione (questo problema viene descritto nel dettaglio in 2.2.4).
- Consente di effettuare analisi di carattere ergonomico quali il calcolo degli indici RULA o REBA che sono basati sugli angoli articolari.

Gli angoli con cui si descrive la posa possono essere espressi in differenti rappresentazioni matematiche quali:

- *Angoli di Eulero:* la rotazione è descritta tramite tre angoli e una sequenza di assi, ad esempio ZXY. Per ottenere la rotazione complessiva bisogna quindi ruotare attorno ad ogni asse secondo l'angolo specificato, nell'ordine dato dalla sequenza. È importante sottolineare che gli angoli di Eulero possono incorrere nel problema del *Gimbal Lock* che comporta la perdita di un grado di libertà, possono quindi non essere la scelta ideale per rappresentare alcune rotazioni del corpo umano.
- *Matrici di rotazione:* sono matrici di dimensione 3×3 che rappresentano la rotazione. Il principale vantaggio di questa forma

è la possibilità di concatenare più rotazioni tramite la moltiplicazione delle relative matrici. Tuttavia, la necessità di utilizzare nove numeri per ogni rotazione, le rende più utili per le fasi di computazione intermedie che per le fasi di predizione: una quantità così elevata di parametri per indicare una sola rotazione implica che questi contengano informazioni ridondanti ed è quindi molto complicato che una rete neurale riesca a generarli in modo che siano tutti coerenti tra loro.

- *Quaternioni:* i quaternioni unitari offrono un metodo pratico di descrivere le rotazioni nello spazio tridimensionale, infatti, sono più compatti e agevoli da manipolare rispetto alle matrici di rotazione, e inoltre non soffrono delle limitazioni associate agli angoli di Eulero. È anche fattibile effettuare moltiplicazioni per combinare rotazioni in successione, seguendo lo stesso principio delle matrici di rotazione. Tuttavia, come in quest'ultime, la presenza di un numero maggiore di parametri per indicare una rotazione che ha soli tre gradi di libertà nello spazio, comporta una certa difficoltà per le reti neurali che devono generare quattro componenti coerenti.

- *Asse-angolo:* questa rappresentazione utilizza un versore per specificare l'asse di rotazione, insieme all'angolo che indica la quantità di cui ruotare. Per ottenere una notazione più compatta, è possibile moltiplicare l'angolo per il versore in modo tale da ridurre da quattro a tre il numero di valori da utilizzare per esprimere ogni rotazione. In questo modo viene inoltre evitato il problema della generazione coerente di tutti i parametri di cui soffrono sia i quaternioni che le matrici di rotazione poichè vengono utilizzati esattamente tre coefficienti per indicare i tre gradi di libertà della rotazione.

La notazione più utilizzata nei dataset considerati è quella tramite asse-angolo grazie alla sua compattezza e robustezza, tuttavia, in letteratura i risultati vengono spesso presentati tramite gli Angoli di Eulero e la relativa metrica (descritta in 3.1.1). È comunque possibile passare facilmente da un'espressione all'altra.

Indipendentemente dal tipo di notazione, la posa di una persona si può esprimere nel seguente modo:

$$x = (e_1, e_2, \cdots, e_J) \in \mathbb{R}^{J \times D} \tag{1}$$

dove J è il numero di giunti dello scheletro considerato ed $e_j \in \mathbb{R}^D$ è la rappresentazione del j-esimo angolo o coordinata, composto da D parametri. Per esempio, nel caso dei quaternioni si ha $D = 4$, mentre

$D = 9$ per le matrici di rotazione. Questa notazione si può estendere ad un'intera sequenza di N pose:

$$X_{1:N} = (x_1, \cdots, x_N) \in \mathbb{R}^{J \times D \times N} \tag{2}$$

dove x_i rappresenta la posa nell'instante $i \in \{1, \cdots, N\}$. Data quindi una sequenza in ingresso $X_{1:N}$ di lunghezza N, il problema della predizione delle successive M pose $\hat{X}_{N+1:N+M}$ si può formulare come segue:

$$\hat{X}_{N+1:N+M} = \arg \max_{X_{N+1:N+M}} P(X_{N+1:N+M} | X_{1:N}) \tag{3}$$

dove $P(X_{N+1:N+M} | X_{1:N})$ è la probabilità che una certa sequenza di M pose segua le N fornite in input. Le reti neurali per la predizione del movimento umano possono essere interpretate come stimatori della probabilità P, data una sequenza di pose in input restituiscono la sequenza futura più probabile. È importante notare che nonostante tale sequenza sia la più probabile tra quelle calcolate, tale probabilità non è nota e potrebbe dunque essere molto bassa: ipotizzando di conoscerla, un esempio limite potrebbe essere un caso in cui di cinque previsioni una sia al 21%, tre siano al 20% e l'ultima al 19%. Di queste cinque previsioni con quasi la stessa probabilità, solo la prima verrà presentata come risultato, questo nonostante i movimenti previsti nei diversi casi potrebbero essere molto diversi.

Le lunghezze N e M sono dei parametri del modello e sono correlate all'orizzonte temporale della previsione dal numero di *frame per secondo* (fps), parametro che indica quanti frame rappresentano un secondo di movimento. In generale, le previsioni fino a 400ms sono definite a *breve termine*, mentre le previsioni su orizzonte temporale maggiore sono definite a *lungo termine*. Ad esempio, se i dati sono espressi a 25fps, i primi 10 degli M frame sono la previsione a breve termine mentre i frame successivi costituiscono la previsione a lungo termine.

1.3.3 *Reti neurali per la predizione del movimento umano*

Di seguito, vengono presentate alcune delle reti neurali più valide nel contesto della previsione del movimento umano, ordinate secondo la data di pubblicazione. Le architetture selezionate per l'analisi in questa tesi verranno poi approfondite nel Capitolo 2.

DMGNN

La rete *Dynamic Multiscale Graph Neural Networks for 3D Skeleton-Based Human Motion Prediction* [10] è una *Graph Neural Network* (GNN) che rappresenta il corpo umano tramite diversi grafi interconnessi tra di

loro in una struttura multilivello. Ciò consente alla rete di apprendere le relazioni che esistono tra le varie componenti del corpo e può essere applicata anche nell'ambito della previsione del movimento umano. Al contrario di altre architetture, DMGNN sfrutta esplicitamente la correlazione tra la posizione delle diverse parti del corpo al fine di prevedere il movimento futuro della persona. Inoltre, le connessioni tra le diverse parti anatomiche non sono predeterminate, vengono infatti apprese durante l'allenamento. Per tali motivi questa rete viene quindi definita *Dinamica* e *Multiscala*. DMGNN è una delle reti neurali scelte per l'analisi in questa tesi per via della sua innovativa struttura basata sui grafi.

RNN-MTP

Il modello di deep learning introdotto in *Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly* [30] è una *Recurrent Neural Network* (RNN) specializzata nella previsione del movimento umano nell'ambito della collaborazione uomo-robot. Il campo d'applicazione di questa rete neurale è pertanto quello che si intende investigare in questa tesi. Tuttavia, le previsioni generate non si concentrano sulla stima della futura posizione di una persona, bensì forniscono un'indicazione generale delle sue azioni, come ad esempio la permanenza in posizione eretta senza interazioni, o la continuazione dell'assemblaggio. A causa di questo approccio meno focalizzato sulla posa della persona, la rete non è stata impiegata nelle analisi successive.

HRI

L'idea chiave alla base dell'architettura *History Repeats Itself: Human Motion Prediction via Motion Attention* [16] è che il movimento umano tende a ripetersi nel corso del tempo. Ciò è particolarmente evidente nelle azioni periodiche, in cui gli stessi schemi di movimento vengono ripetuti continuamente ad una certa frequenza. Questa interpretazione vale anche per movimenti più complessi in cui i singoli sottomovimenti sono molto più numerosi e generalmente ripetuti a intervalli di tempo maggiori. Per trovare informazioni storiche rilevanti, HRI implementa un meccanismo di *attenzione* del movimento che mira a riconoscere queste sotto-azioni avvenute in diversi momenti. Queste informazioni, combinate con la sequenza di movimenti passata, vengono elaborate da una *Graph Convolution Network* (GCN) per apprendere le relazioni tra le diverse articolazioni dello scheletro umano, che vengono poi utilizzate per prevedere le pose future. HRI è una delle reti neurali scelte per l'analisi in questa tesi per via dell'utilizzo del meccanismo dell'*attenzione* e perchè codifica le informazioni temporali nel dominio della frequenza tramite la *Discrete Cosine Transform* (DCT).

LPJP

Anche in *Learning Progressive Joint Propagation for Human Motion Prediction* [3] la previsione del movimento umano viene effettuata sfruttando l'architettura *Transformer* e quindi il meccanismo dell'attenzione. I dati relativi alle posizioni dei giunti vengono così codificati sia dal punto di vista spaziale che temporale. In seguito, per ottenere una migliore comprensione della posa della persona, viene utilizzata una strategia di codifica progressiva che lavora in modo da analizzare prima le parti centrali del corpo per poi estendere verso le parti più periferiche. Infine, gli schemi di movimento appresi durante l'addestramento vengono salvati in un dizionario che funge da memoria in modo da guidare le previsioni durante l'utilizzo della rete.

PVRED

A Position-Velocity Recurrent Encoder-Decoder for Human Motion Prediction [29] è un modello di deep learning basato su un'architettura RNN. Gli approcci alla previsione del movimento umano con RNN standard, il cui input è la sola sequenza di pose, tendono a convergere in una posa statica dopo alcuni istanti di previsione o non riescono a generare sequenze dall'aspetto naturale. Per superare questo problema, migliorando così le previsioni a lungo termine, gli autori hanno proposto una struttura che incorpora anche informazioni sulla velocità della posa e una codifica delle relazioni temporali tra i vari frame della sequenza di movimento. La parametrizzazione e l'allenamento della rete tramite quaternioni al posto della notazione asse-angolo consente inoltre di evitare problemi legati a singolarità e discontinuità. PVRED è stata selezionata come una delle reti neurali per l'analisi in questa tesi, in quanto si tratta di una RNN basata su una *Gated Recurrent Unit* (GRU), che è stata ampiamente utilizzata in passato per la previsione del movimento.

STT

L'architettura *A Spatio-temporal Transformer for 3D Human Motion Prediction* [2] si basa sull'impiego dei *Transformer*, utilizzando il meccanismo dell'*attenzione* per estrarre rappresentazioni spazio-temporali e generare pose su orizzonti temporali sia a breve che a lungo termine. In questa rete neurale il meccanismo dell'attenzione ha un duplice utilizzo: dal punto di vista *temporale* viene sfruttato per ricavare le relazioni tra le diverse posizioni di un'articolazione al variare del tempo; dal punto di vista *spaziale* l'attenzione viene adoperata per ricavare le informazioni sulle reciproche relazioni dei giunti nello stesso frame. Questo modello consente quindi di ottenere delle accurate predizioni a breve termine e di genereare delle sequenze di movimento plausibili per intervalli di previsione più estesi. L'impiego evoluto dell'architettura *Transformer* rende STT una rete molto interessante.

DMMGAN

Anche il modello proposto in *Diverse Multi Motion Prediction of 3D Human Joints using Attention-Based Generative Adverserial Network* [21] si basa sull'impiego dei *Transformer*, tuttavia si distingue dagli altri approcci di previsione del movimento umano in diversi modi. Innanzitutto, a differenza di molti altri modelli che si limitano a predire un singolo movimento, questa rete genera un'intera gamma di scenari futuri possibili. Inoltre, non si limita a prevedere la posa del soggetto, ma fornisce anche il suo movimento nell'ambiente circostante. L'impostazione multi-futuro di questa rete la rende al contempo di notevole interesse ma anche di difficile comparazione con altre architetture, pertanto non è stata utilizzata nell'ambito di questa tesi.

BiTGAN

L'architettura introdotta in *Bidirectional Transformer GAN for Long-term Human Motion Prediction* [31] è forse una delle più affascinanti nell'ambito della predizione del movimento umano. L'idea alla base consiste in una rete che consente di predire le pose future basandosi su quelle passate ma, allo stesso tempo, è possibile ricavare le pose passate partendo da quelle predette per il futuro. Questa impostazione bidirezionale è un fattore fondamentale che porta a delle pose future plausibili, poiché da queste si può ritornare alle pose originali che le hanno generate e verificarne la somiglianza. Grazie a questa strategia, BiTGAN raggiunge degli ottimi risultati nelle previsioni a lungo termine. Purtroppo, questa rete neurale non è stata utilizzata a causa dell'indisponibilità del codice sorgente, che rende impossibile l'utilizzo della stessa nell'ambito di questa tesi.

2

ARCHITETTURE PER LA PREVISIONE DEL MOVIMENTO

In questo capitolo vengono esaminate nel dettaglio le reti DMGNN, HRI e PVRED, selezionate per l'analisi in questa tesi tra quelle proposte in 1.3.3. Per ogni rete viene descritto il principio di funzionamento, i dettagli dell'allenamento e le modifiche implementate al fine di compararne i risultati tramite le metriche e i dataset desiderati.

2.1 DMGNN

Dynamic Multiscale Graph Neural Networks for 3D Skeleton-Based Human Motion Prediction [10] è una rete neurale che si basa su una rappresentazione del corpo umano tramite grafi di diversa dimensione interconnessi tra di loro in una struttura multilivello.

Una *Graph Neural Network* (GNN) è un tipo di rete che elabora dati strutturati nella forma di grafi, cioè un insieme di nodi e di archi che li collegano. Questa struttura consente alla rete di apprendere le relazioni che esistono tra i nodi e può essere applicata anche nell'ambito della previsione del movimento umano. Infatti, il posizionamento relativo di diverse parti del corpo possiede un'informazione cruciale per prevedere il comportamento di una persona negli istanti immediatamente succesivi, tuttavia, al contrario della rete sotto analisi, molte ricerche in questo campo non sfruttano esplicitamente questa correlazione, rendendo questo un approccio particolarmente cruciale da investigare.

In DMGNN il corpo di una persona viene rappresentato tramite una serie di grafi con una quantità di nodi che decresce da un livello al successivo. La rete proposta è costituita da un modello con tre livelli così strutturati: il grafo di livello più basso utilizza un nodo per ogni giunto del corpo, il grafo intermedio ha due nodi per arto e altri due che modellano rispettivamente testa e torso, infine, il grafo al livello superiore utilizza un nodo per ogni arto più uno relativo a testa e torso per un totale di soli cinque nodi.

I grafi dei singoli livelli (*single-scale graph*) vengono poi interconessi tra loro da degli ulteriori grafi (*cross-scale graph*) che collegano i nodi di diversi livelli. Per esemplificare, il nodo di un arto superiore (livello 3) può collegarsi con braccio e avambraccio dello stesso lato (livello 2), l'avambraccio a sua volta potrebbe essere collegato con mano e polso (livello 1). I pesi di ogni connessione non sono predeterminati, vengono infatti appresi durante l'allenamento della rete. Per tali motivi questa GNN viene quindi definita *Dinamica* e *Multiscala*.

A livello macroscopico, l'architettura di DMGNN è composta da un *Encoder* che elabora le pose in ingresso e le fornisce ad un *Decoder* che effettua la previsione. Questi blocchi vengono di seguito descritti nel dettaglio, inoltre, in Figura 3 si può osservare l'architettura generale di DMGNN.

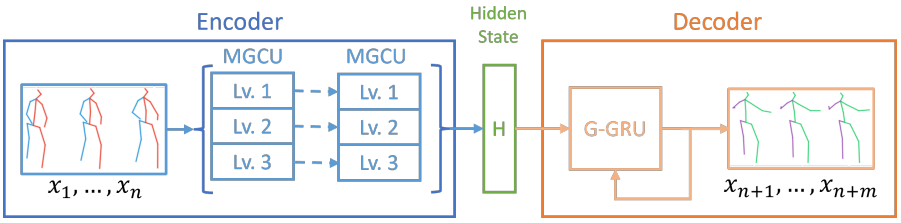


Figura 3: Schema semplificato dell'architettura di DMGNN, basata su un sistema Encoder-Decoder. L'Encoder è formato da una serie di blocchi chiamati *Multiscale Graph Computational Unit* (MGCU) che elaborano le pose passate mentre il Decoder è composto da una *Graph-based Gate Recurrent Unit* (G-GRU) che prevede le pose future in modo ricorsivo.

2.1.1 *Principio di funzionamento dell'Encoder*

L'Encoder si occupa di elaborare le pose passate in modo da ottenerne una rappresentazione da fornire in ingresso al Decoder. È formato da una serie di blocchi in cascata chiamati *Multiscale Graph Computational Unit* (MGCU) il cui compito consiste nell'estrarre e nel fondere insieme le informazioni dai diversi livelli con cui viene rappresentato il corpo umano.

Grazie alla dinamicità della rete in fase di addestramento, ogni blocco ha al suo interno una struttura dei grafi differente dalle altre unità, consentendo in questo modo l'apprendimento di un'ampia varietà di schemi di correlazione delle articolazioni.

Una MGCU include due tipi di componenti fondamentali, un *Single-Scale graph convolution Block* (SSB) e un *Cross-Scale fusion Block* (CSB) che si occupano rispettivamente di estrarre le informazioni da un singolo livello e di fonderle tra livelli adiacenti. In una MGCU con n livelli ci sono quindi n SSB e $n-1$ CSB.

In un SSB, per estrarre le informazioni relative alla posa e al movimento, si utilizzano operazioni di convoluzione sia spaziale che temporale. Definendo con M_s il numero di componenti del corpo nel livello s e con D la dimensione di ogni componente, l'equazione della convoluzione spaziale di un SSB è:

$$X_{s,sp} = \sigma(A_s X_s W_s + X_s U_s) \in \mathbb{R}^{M_s \times D'} \tag{4}$$

dove $X_s \in \mathbb{R}^{M_s \times D}$ è l'input dal livello s, $W_s, U_s \in \mathbb{R}^{D \times D'}$ sono i pesi da allenare, $\sigma(\cdot)$ è la funzione non lineare *Rectified Linear Unit*

(ReLU) [20] ed infine $A_s \in \mathbb{R}^{M_s \times M_s}$ è la matrice di adiacenza addestrabile che rappresenta le connessioni dei nodi del grafo. A_s viene inizializzata con dei valori che rappresentano le connessioni fisiche tra le parti del corpo umano ma, durante l'allenamento della rete, può apprendere in modo flessibile delle nuove correlazioni tra i dati. Grazie a ciò, nella rete addestrata, ogni SSB avrà una diversa matrice di adiacenza che rappresenta un diverso insieme di connessioni. Dopodiché, per ricavare le correlazioni tra i diversi istanti del movimento, viene effettuata una convoluzione temporale, la cui formulazione è analoga a quanto riportato per la convoluzione spaziale.

Il compito di diffondere le informazioni tra i diversi livelli di una MGCU è affidato ai CSB, ognuno di questi composto da un tipo di grafo detto bipartito, cioè un grafo in cui i nodi sono suddivisi in due insiemi e ciascun arco collega un nodo di un insieme ad uno dell'altro. Queste connessioni tra diversi livelli vengono apprese in modo dinamico durante l'addestramento della rete.

Come esempio si consideri il CSB dal livello s_1 a s_2, la matrice di adiacenza che rappresenta il grafo sarà $A_{s_1 s_2} \in [0,1]^{M_{s_2} \times M_{s_1}}$. Si indichi con $p_{s_1,i}$ e $p_{s_2,k}$ la rappresentazione in vettori della convoluzione temporale dell'i-esimo giunto e della k-esima parte. La forza della connessione tra questi due nodi si ottiene nel seguente modo:

$$\begin{aligned}
r_{s_1,i} &= \sum_{j=1}^{M_{s_1}} f_{s_1}\left(\left[p_{s_1,i}, p_{s_1,j} - p_{s_1,i}\right]\right) \\
h_{s_1,i} &= g_{s_1}\left(\left[p_{s_1,i}, r_{s_1,i}\right]\right) \\
r_{s_2,k} &= \sum_{j=1}^{M_{s_2}} f_{s_2}\left(\left[p_{s_2,k}, p_{s_2,j} - p_{s_2,k}\right]\right) \\
h_{s_2,k} &= g_{s_2}\left(\left[p_{s_2,k}, r_{s_2,k}\right]\right) \\
\left(A_{s_1 s_2}\right)_{k,i} &= \text{softmax}\left(h_{s_2,k}^{\top} h_{s_1,i}\right) \in [0,1]
\end{aligned} \tag{5}$$

dove $f_{s_1}(\cdot)$, $g_{s_1}(\cdot)$, $f_{s_2}(\cdot)$, $g_{s_2}(\cdot)$ sono delle reti *Multi-Layer Perceptron* (MLP) e $[\cdot,\cdot]$ è la concatenazione.

In seguito, il risultato del CSB viene utilizzato per aggiornare il risultato del SSB:

$$X_{s_2} \leftarrow A_{s_1 s_2} X_{s_1} W_{F,s_1} + X_{s_2} \in \mathbb{R}^{M_{s_2} \times D} \tag{6}$$

in cui $W_{F,s_1} \in \mathbb{R}^{D \times D}$ è una matrice di pesi addestrabile.

Il valore X_{s_2} ottenuto verrà poi usato come input per il SSB del secondo livello della MGCU successiva. Allo stesso modo si possono definire le operazioni dal livello s_2 a s_1 e tra i livelli s_2 e s_3.

Il risultato ottenuto dall'ultima MGCU viene infine combinato tramite una media pesata di ogni livello e una media sull'asse temporale, si aggrega così l'informazione delle pose passate in un unico embedding H che diventerà l'ingresso per il Decoder.

2.1.2 *Principio di funzionamento del Decoder*

Il Decoder è composto da una *Graph-based Gate Recurrent Unit* (G-GRU) che prevede le pose future in modo ricorsivo sotto la guida di un grafo, fattore che consente di regolarizzare gli stati utilizzati per generare le pose future. Una GRU è un tipo di RNN e come tale ha la particolarità di mantenere una memoria interna (l'*hidden state*, indicato con $H^{(t)}$ per l'istante t) che ad ogni passo di previsione viene aggiornata e utilizzata per produrre la posa per quell'istante. Per il primo istante di previsione, l'hidden state è inizializzato tramite l'embedding H prodotto dall'Encoder. La scelta di una GRU è dovuta alla minor suscettibilità al problema della scomparsa dei gradienti e alle minori risorse computazionali richieste per l'addestramento rispetto ad altri tipi di RNN.

L'aggiornamento del Decoder si basa sulla matrice di adiacenza $A_H \in \mathbb{R}^{M\times M}$ che rappresenta il grafo, inizializzata basandosi sullo scheletro reale. Data l'informazione $X^{(t)} \in \mathbb{R}^{M\times d}$ sulla posa predetta per l'istante t e l'hidden state della G-GRU $H^{(t)} \in \mathbb{R}^{M\times D_h}$, l'hidden state per l'istante successivo viene calcolato nel seguente modo:

$$\begin{aligned} r^{(t)} &= \sigma\left(r_{in}\left(X^{(t)}\right) + r_{hid}\left(A_H H^{(t)} W_H\right)\right), \\ u^{(t)} &= \sigma\left(u_{in}\left(X^{(t)}\right) + u_{hid}\left(A_H H^{(t)} W_H\right)\right), \\ c^{(t)} &= \tanh\left(c_{in}\left(X^{(t)}\right) + r^{(t)} \odot c_{hid}\left(A_H H^{(t)} W_H\right)\right), \\ H^{(t+1)} &= u^{(t)} \odot H^{(t)} + \left(1 - u^{(t)}\right) \odot c^{(t)}, \end{aligned} \tag{7}$$

dove W_H indica la matrice dei pesi e $r_{in}(\cdot)$, $r_{hid}(\cdot)$, $u_{in}(\cdot)$, $u_{hid}(\cdot)$, $c_{in}(\cdot)$, $c_{hid}(\cdot)$ sono delle mappe lineari addestrabili.

Dato un istante t, la posa $\hat{X}^{(t+1)}$ per il passo successivo si ottiene nel seguente modo:

$$\hat{X}^{(t+1)} = \hat{X}^{(t)} + f_{pred}\left(\text{G-GRU}\left(\hat{X}^{(t)}, H^{(t)}\right)\right) \tag{8}$$

in cui $\hat{X}^{(t)}$ è la posa predetta per il passo t, $H^{(t)}$ è l'hidden state, G-GRU$(\cdot)$ è la funzione realizzata dalla gate recurrent unit e $f_{pred}(\cdot)$ è una funzione d'uscita realizzata tramite una rete MLP. Dall'equazione 8 si osserva quindi che la posa per il frame successivo si ottiene dal Decoder aggiungendo alla posa del passo precedente la differenza stimata.

2.1.3 *Allenamento della rete*

L'addestramento della rete è stato effettuato considerando come *loss function* la distanza l_1 (anche conosciuta come *distanza di Manhattan*) tra i vettori rappresentanti le pose previste e quelle reali. Traducendo in formule si ottiene:

$$L_{PRED} = \frac{1}{N} \sum_{n=1}^{N} \left\| \hat{X}_n - X_n \right\|_1 \tag{9}$$

in cui $\hat{X}_n, X_n$ sono rispettivamente le pose previste e reali nell'istante n. La distanza l_1 viene preferita alla più classica l_2 (*distanza Euclidea*) poiché empiricamente ha portato a risultati migliori in termini di errore ottenuto. Infatti, consente di mitigare l'esplosione del gradiente nei giunti con funzione di perdita molto grande e permette al contempo di non lasciar sparire il gradiente nei giunti in cui la perdita è molto bassa.

La rete DMGNN è stata allenata dai suoi autori sui dataset Human3.6M e CMU (vedi 1.3.1) utilizzando come input i soli giunti in notazione asse-angolo il cui valore varia. Ciò non vale per tutti i giunti, infatti i valori angolari per polsi e caviglie spesso non vengono riportati nei dataset e sono quindi lasciati a zero. La frequenza dei dati viene inoltre ridotta a 25 fps portando quindi la distanza tra un frame e il successivo a 40 ms.

2.1.4 *Modifiche implementate*

Dei dataset utilizzati nell'analisi effettuata in questa tesi (H3.6M [8] e AMASS [14]) solo H3.6M è stato impiegto dagli autori di DMGNN. Pertanto, è stato necessario apportare delle modifiche all'implementazione originale al fine di consentire la compatibilità con la differente rappresentazione del corpo umano presente in AMASS. Nello specifico, queste modifiche hanno comportato l'integrazione della sequenza di operazioni necessarie per caricare correttamente il dataset stesso. Inoltre, è stato necessario definire i grafi iniziali per ciascuno dei tre livelli delle MGCU dell'Encoder, determinando così il raggruppamento più appropriato per i diversi giunti dello scheletro. Oltre a ciò, sono state introdotte alcune metriche, come descritto nella sezione 3.1, al fine di consentire un confronto tra i risultati ottenuti da DMGNN e quelli ottenuti dalle altre reti neurali considerate.

2.2 HRI

History Repeats Itself: Human Motion Prediction via Motion Attention [16] è una rete neurale feed-forward che si basa sul meccanismo dell'attenzione. L'ipotesi su cui si fonda questa soluzione è la ciclicità del

movimento umano: i movimenti effettuati nel passato probabilmente si ripeteranno in futuro. Naturalmente, il funzionamento di questa architettura è tanto migliore quanto più il movimento da prevedere soddisfa questa ipotesi di ripetitività, degli esempi possono essere camminare, correre o salutare una persona, cioè una serie di brevissime azioni che si ripetono con grande frequenza. Tuttavia, se il movimento è non periodico, ad esempio sedersi su una sedia o fare una foto, la previsione effettuata sarà chiaramente meno accurata. Nonostante ciò, movimenti all'apparenza non periodici possono risultare in realtà delle lunghe serie di micro-movimenti che si ripetono dopo grandi intervalli di tempo.

Macroscopicamente il funzionamento della rete si può dividere in due parti distinte:

- un *Modello di Attenzione del Movimento* in cui vengono aggregate le informazioni della storia passata del movimento della persona per formare una stima del movimento futuro,
- un *Modello di Previsione* che combina il risultato dell'attenzione assieme all'ultima sottosequenza osservata per ricavare le dipendenze spaziali e temporali dei dati ed ottenere così la previsione del movimento.

Viene ora analizzato nel dettaglio il funzionamento di questi due blocchi. Si indica con $X_{1:N} = [x_1, x_2, x_3, \cdots, x_N]$ la storia passata del movimento, formata da N pose x_i, dove $x_i \in \mathbb{R}^{K \times D}$ con K il numero di giunti che descrivono la posa e D il numero di parametri che descrive ogni giunto (ad esempio $D = 3$ per la notazione asse-angolo). L'obiettivo è predire i futuri T istanti ed ottenere le pose $X_{N+1:N+T}$.

2.2.1 *Modello di Attenzione del Movimento*

Mantenendo il formalismo introdotto nell'articolo *Attention Is All You Need* [27] il modello dell'attenzione può essere descritto come una mappatura da una *query* e un insieme di coppie *key-value* ad un output. Quest'ultimo è una somma pesata dei *values* in cui i pesi (cioè l'*attenzione*) sono calcolati in funzione della *query* e della corrispondente *key*.

Nel caso del modello di attenzione del movimento la storia passata viene divisa in sequenze di $M + T$ pose. Per ogni sequenza, la sottosequenza composta dalle prime M pose corrisponde alla *key* mentre il *value* è l'intera sequenza di lunghezza $M + T$. La *query* non è altro che l'ultima sequenza di M pose della storia passata di cui si vuole prevedere le future T pose.

Il numero di sequenze estratte dalla storia è $N + 1 - (M + T)$, in Figura 4 si può vedere un esempio numerico.

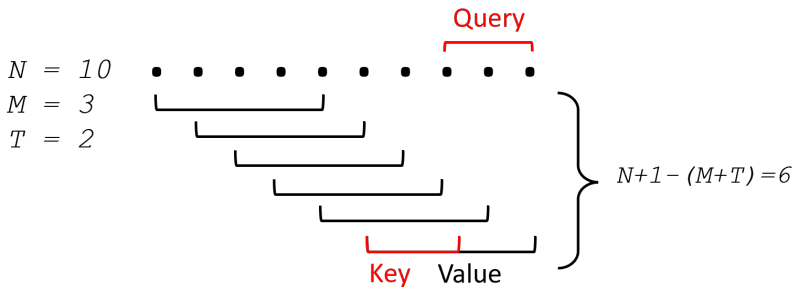


Figura 4: Esempio numerico dell'estrazione di *key* (M pose), *value* (M + T pose) e *query* (M pose) dalla storia passata (N pose).

I *values* vengono poi mappati nel dominio delle frequenze tramite la *Trasformata Discreta del Coseno* (DCT). Data una sequenza di coordinate (o angoli) di un giunto $\{x_l\}_{l=1}^{L}$, i suoi coefficienti DCT $\{C_l\}_{l=1}^{L}$ si calcolano con:

$$C_l = \sqrt{\frac{2}{L}} \sum_{n=1}^{L} x_n \frac{1}{\sqrt{1+\delta_{l1}}} \cos\left(\frac{\pi}{L}\left(n-\frac{1}{2}\right)(l-1)\right) \tag{10}$$

dove $l, n \in \{1, 2, \cdots, L\}$ e $\delta_{i,j} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$

L'operazione inversa, ovvero la *Trasformata Discreta del Coseno Inversa* (IDCT), si effettua in modo analogo:

$$x_n = \sqrt{\frac{2}{L}} \sum_{l=1}^{L} C_l \frac{1}{\sqrt{1+\delta_{l1}}} \cos\left(\frac{\pi}{L}\left(n-\frac{1}{2}\right)(l-1)\right) \tag{11}$$

I coefficienti ottenuti dalla DCT sono $V_i \in \mathbb{R}^{K\times(M+T)}$ e a questi ci si riferirà parlando di *values* nei passaggi successivi. In ogni riga di V_i ci sono quindi i coefficienti della DCT di un singolo giunto.

La *query* e le *keys* vengono invece mappate in vettori di dimensione d da due funzioni $f_q : \mathbb{R}^{K\times M} \to \mathbb{R}^d$ e $f_k : \mathbb{R}^{K\times M} \to \mathbb{R}^d$ modellate tramite rete neurale. Si può quindi scrivere:

$$q = f_q\left(X_{N+1-M:N}\right), \quad k_i = f_k\left(X_{i:i+M-1}\right) \tag{12}$$

dove $q, k_i \in \mathbb{R}^d$, e $i \in 1, 2, \cdots, N+1-(M+T)$. Le funzioni f_q e f_k utilizzano come non linearità la funzione ReLU in modo tale che q e k_i siano non negativi. L'attenzione relativa ad ogni *key* viene poi calcolata nel seguente modo:

$$a_i = \frac{q k_i^T}{\sum_{t=1}^{N+1-(M+T)} q k_i^T} \tag{13}$$

Invece della funzione *softmax*, comunemente usata nei meccanismi basati sull'attenzione, i punteggi ottenuti dal prodotto tra la *query* e le *keys* vengono soltanto normalizzati, evitando così il problema della scomparsa del gradiente. Il precedente utilizzo della funzione ReLU consente di ottenere dei valori di attenzione non negativi, inoltre, la normalizzazione impedisce a tali valori di essere superiori ad uno. Complessivamente si ottiene quindi $a_i \in [0, 1]$.

L'output del modello di attenzione è in seguito calcolato come somma pesata dei *values*:

$$U = \sum_{t=1}^{N+1-(M+T)} a_i V_i \tag{14}$$

dove $U \in \mathbb{R}^{K\times(M+T)}$. Questa stima iniziale viene utilizzata dal modello di previsione descritto successivamente per generare la stima delle pose future $\hat{X}_{N+1:N+T}$. Per ottenere una previsione su un orizzonte temporale più lungo è possibile utilizzare le T pose predette come ulteriore input alla rete per prevedere altre T pose, continuando in questo modo fino ad ottenere la sequenza di pose di lunghezza desiderata.

2.2.2 *Modello di Previsione*

La rappresentazione tramite DCT consente di codificare le relazioni temporali di come si muove ogni giunto nelle sequenze. Oltre a ciò, per ottenere una previsione accurata, è necessario ricavare le relazioni spaziali tra le articolazioni del corpo. In sostanza, si tratta di capire come si muove una parte del corpo in relazione ad una parte differente. Per far ciò viene utilizzata una *Graph Convolution Network* (GCN) di tipo completamente connessa con K nodi rappresentanti i K giunti del corpo.

L'input al layer p è una matrice $H^{(p)} \in \mathbb{R}^{K\times F}$ in cui ogni riga relativa ad uno dei K nodi è il vettore F dimensionale delle caratteristiche. Per esempio, l'input al primo layer è $H^{(1)} \in \mathbb{R}^{K\times 2(M+T)}$ che è la concatenazione della matrice U definita in (14) e un'altra matrice $D \in \mathbb{R}^{K\times(M+T)}$. Questa matrice D viene ricavata applicando la DCT ad una sequenza di $M + T$ pose di cui le prime M sono la sequenza $X_{N-M+1:N}$, cioè la *query*, mentre le ultime T pose sono una replica dell'ultima posa X_N.

Ogni layer p della GCN produce un output $H^{(p+1)} \in \mathbb{R}^{K\times\hat{F}}$, partendo dal risultato del layer precedente $H^{(p)}$ secondo l' equazione:

$$H^{(p+1)} = \sigma\left(A^{(p)}H^{(p)}W^{(p)}\right), \tag{15}$$

dove $A^{(p)} \in \mathbb{R}^{K\times K}$ è la matrice di adiacenza addestrabile del layer p, rappresentante la forza della connessione tra i nodi, $W^{(p)} \in \mathbb{R}^{F\times\hat{F}}$

contiene altri pesi usati per estrarre le caratteristiche e $\sigma(\cdot)$ è una funzione di attivazione, in questo caso $\tanh(\cdot)$. La GCN complessiva è infine realizzata da un totale di 12 blocchi, ognuno contenente due di questi layer, più un layer iniziale per mappare i coefficienti DCT e uno finale per riottenere i coefficienti DCT dalla rete.

Applicando alla matrice ottenuta dall'ultimo layer della GCN la IDCT è possibile riottenere una sequenza di lunghezza $M + T$ le cui ultime T pose rappresentano la predizione $\hat{X}_{N+1:N+T}$.

2.2.3 *Allenamento della rete*

La rete è stata allenata dai suoi autori sui dataset Human3.6M e AMASS (vedi 1.3.1) utilizzando per entrambi la rappresentazione della posa tramite posizioni assolute 3D dei giunti del corpo.

Oltre a testare il funzionamento sui rispettivi set di test, per dimostrare la capacità di generalizzare di questa architettura la rete basata su AMASS è stata inoltre valutata sul dataset 3DPW.

In aggiunta, è stata addestrata una terza rete utilizzando la rappresentazione angolare del dataset Human3.6M, al fine di consentire un confronto diretto con altre ricerche in questo settore.

Le reti basate sulle posizioni assolute sono state allenate utilizzando come *loss function* l'errore di posizione medio per giunto l_{MPJPE} (dall'inglese *Mean Per Joint Position Error*) così definito:

$$L_{MPJPE} = \frac{1}{J(M+T)} \sum_{t=1}^{M+T} \sum_{j=1}^{J} \left\| \hat{p}_{t,j} - p_{t,j} \right\|_2 \tag{16}$$

dove $p_{t,j}, \hat{p}_{t,j} \in \mathbb{R}^3$ sono la posizione reale e stimata del giunto j al tempo t.

Per la rete basata sulla rappresentazione in angoli la loss function utilizzata è la distanza media l_1 la cui è formula è comunque molto simile a quella precedente:

$$L_{ANG} = \frac{1}{K(M+T)} \sum_{t=1}^{M+T} \sum_{k=1}^{K} \left\| \hat{\alpha}_{t,k} - \alpha_{t,k} \right\|_1 \tag{17}$$

dove $\alpha_{t,k}$ e $\hat{\alpha}_{t,k}$ sono l'angolo k reale e stimato al tempo t.

2.2.4 *Modifiche implementate*

Le reti implemetate dagli autori sono state allenate sugli stessi dataset che si intende utilizzare per l'analisi in questa tesi. Considerando che i risultati dell'allenamento sono liberamente disponibili online, non è stato quindi necessario riallenarle.

Tuttavia, la rete allenata su AMASS è basata sulle posizioni assolute dei giunti, mentre alcune delle metriche implementate per valutare la bontà delle previsioni (descritte in sezione 3.1) si basano su una rappresentazione angolare. Si è quindi reso necessario implementare un sistema di cinematica inversa che consentisse di risalire agli angoli tra i giunti partendo dalle loro posizioni assolute.

Considerando la catena cinematica dei giunti, dato un giunto j definiamo p il giunto padre e f il giunto figlio. La soluzione implementata consiste nel calcolare per ogni giunto j l'angolo formato dalle posizioni dei giunti p, j, f. Ad esempio, facendo riferimento alla Figura 1, per calcolare l'angolo formato dal gomito destro è necessario conoscere la posizione della spalla (giunto 17), del gomito (giunto 19) e del polso (giunto 21).

Ovviamente, questa soluzione non si può applicare per i giunti senza padre, ovvero per il bacino, o per i giunti senza figlio, ovvero i giunti terminali come le mani, i piedi e la testa. Questa limitazione in realtà non costituisce un problema, poichè in questi dataset l'angolo associato al bacino contiene la rotazione rispetto al sistema di riferimento globale e non è quindi importante al fine di conoscere la posa della persona. Per quanto riguarda gli angoli relativi a mani e piedi, in letteratura questi vengono spesso scartati quando si calcola la qualità della previsione e non sono quindi fondamentali.

Altro caso particolare è il giunto della schiena a cui sono collegati i giunti delle spalle e del collo. La presenza di tre giunti figli non è problematica, poichè, per come sono implementate le catene cinematiche di AMASS e Human3.6M, il primo giunto delle spalle è in posizione fissa rispetto alla schiena e quindi il giunto figlio da scegliere per il calcolo dell'angolo è quello del collo.

L'unica vera limitazione di questo approccio è dovuta ad una leggera perdita di informazione dovuta all'impossibilità di ricostruire fedelmente la posa della persona a partire dalle sole posizioni assolute dei giunti. Infatti, alcune rotazioni del corpo non producono spostamenti delle posizioni dei giunti, si consideri ad esempio la posizione della mano con il palmo rivolto verso l'alto o verso il basso. Questa rotazione di polso e gomito non si riflette in alcun modo in uno spostamento della posizione del polso e non è quindi ricavabile considerando solo quest'ultima. Per questo motivo, i sistemi di cattura ottici con cui vengono registrati questo tipo di dataset richiedono la presenza di almeno due marcatori per giunto, in modo da poter ricostruire tutte le rotazioni del corpo. Nonostante ciò, la perdita di informazione è di entità trascurabile.

Inoltre, il calcolo della cinematica inversa viene effettuato al solo scopo di testare la qualità della rete sotto esame. I risultati non vengono quindi usati così come sono ma vengono paragonati agli angoli reali, ottenuti anch'essi tramite lo stesso algoritmo a partire dalle po-

sizioni reali dei giunti. Le conclusioni derivate dalle metriche non risultano quindi viziate dalle restrizioni di questo metodo.

2.3 PVRED

A Position-Velocity Recurrent Encoder-Decoder for Human Motion Prediction [29] è un tipo di *Recurrent Neural Network* (RNN) appositamente concepita per anticipare i movimenti umani. Questa previsione avviene tenendo conto non solo delle pose più recenti della persona, ma anche delle relazioni temporali tra queste pose e delle loro velocità. Grazie a questi accorgimenti, le pose previste risultano coerenti e si elimina il problema di convergenza verso la posizione media, un inconveniente comune in molte altre soluzioni basate su RNN per cui la posizione prevista tende a convergere rapidamente verso la posizione media del dataset di addestramento, spesso già entro poche decine di frame. Un'altra particolarità di questa rete è l'utilizzo di una parametrizzazione degli angoli dei giunti in output tramite quaternioni, consentendo così di ottenere una previsione priva di singolarità e di discontinuità.

L'architettura PVRED è un'evoluzione di RED (*Recurrent Encoder-Decoder* [18]) un'altra architettura basata su RNN sviluppata per la predizione del movimento umano. Di seguito viene quindi descritto il funzionamento di RED e le migliorie apportate da PVRED.

2.3.1 *Principio di funzionamento di RED*

L'architettura di RED è basata su RNN, un tipo di rete neurale che opera su una sequenza in input di lunghezza variabile e che, tramite una sorta di memoria interna, l'*hidden state*, consente di utilizzare le informazioni dagli input precedenti per influenzare l'output attuale. Data una sequenza in ingresso $X = (x_1, \cdots, x_n)$ dove x_t è l'input all'istante t, l'hidden state è $h_t = f(h_{t-1}, x_t)$ con f funzione di attivazione non lineare.

Una RNN standard soffre del problema della sparizione del gradiente, un fenomeno che si verifica durante l'addestramento quando i gradienti che vengono calcolati diventano quasi nulli. In questa situazione, l'aggiornamento dei pesi tramite la tecnica della discesa del gradiente diventa inefficace e la rete fatica ad apprendere.

Per ovviare al problema, sono state sviluppate diverse tecniche tra cui l'uso di particolari funzioni di attivazione, l'inizializzazione intelligente dei pesi e l'utilizzo di architetture specifiche come le *Gated Recurrent Unit* (GRU) e le *Long Short-Term Memory* (LSTM). RED impiega le GRU perchè consentono di ottenere una migliore efficienza computazionale rispetto alle LSTM.

La struttura di RED è composta da due RNN chiamate *Encoder* e *Decoder*. Il compito dell'Encoder è trasformare la sequenza di po-

se $X = (x_1, \cdots, x_n)$ in ingresso in un vettore di lunghezza fissa da dare come input al Decoder. Questo vettore è l'ultimo hidden state dell'Encoder h_n^E, ed è utilizzato come hidden state iniziale h_0^D per il Decoder. Ogni posa y_t della sequenza predetta $Y = (y_1, \cdots, y_m)$ si ottiene dall'hidden state del Decoder h_t^D secondo l'equazione:

$$y_t = y_{t-1} + W h_{t-1}^D + b \tag{18}$$

in cui y_{t-1} è la posa prevista al passo precedente oppure, per il primo istante di previsione ($t = 1$), l'ultima posa della sequenza in input (cioè x_n). Infine, W e b sono rispettivamente i pesi e i parametri di bias appresi durante l'addestramento della rete.

2.3.2 *Principio di funzionamento di PVRED*

La struttura generale di PVRED, schematizzata in Figura 5, è formata da due RNN come in RED, tuttavia sia l'Encoder che il Decoder hanno una maggiore complessità. In ingresso all'Encoder, oltre alla sequenza di pose, vengono fornite anche le velocità delle pose stesse (la velocità di rotazione di ogni giunto del corpo) e una codifica della posizione relativa delle varie pose (indicata con il termine inglese *position embedding*).

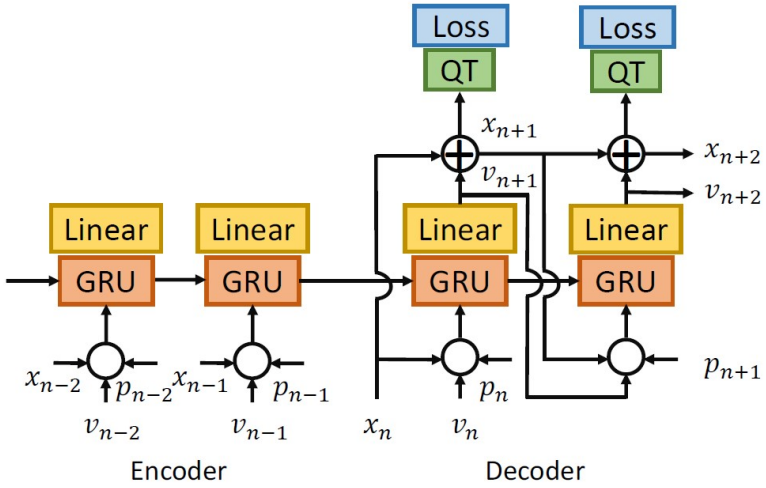


Figura 5: Schema del funzionamento di PVRED, tratto da [29]. Sia l'Encoder che il Decoder hanno in ingresso le pose x_n, le velocità v_n e le posizioni p_n. QT indica il layer dove avviene la trasformazione da asse-angolo a quaternioni.

L'inclusione della velocità nei dati in ingresso è giustificata dal contributo portato nel preservare la continuità del movimento. Per ogni istante t, la velocità v_t di una posa x_t viene espressa semplicemente come la differenza tra gli istanti t e $t-1$, quindi $v_t = x_t - x_{t-1}$.

L'altro parametro, la codifica della posizione, mira a determinare la relazione temporale tra pose in momenti distinti, allo scopo di identificare corrispondenze tra di esse. Questo input aggiuntivo riveste notevole importanza, poiché contribuisce a prevenire la convergenza della posizione predetta verso la posa media anche in scenari di previsione a lungo termine, come quelli superiori al secondo. Il metodo più semplice per ottenere il positional embedding di una specifica posa all'istante t è utilizzare un vettore di zeri con un singolo valore pari a uno in posizione t-esima. Tuttavia, poiché questo approccio non offre la flessibilità necessaria per gestire sequenze di input di lunghezza variabile, in PVRED si ricorre a funzioni seno e coseno di diverse frequenze.

Data una sequenza di pose in input composta da n frame, se si vuole predire una sequenza di lunghezza m, la codifica di posizione per il frame t, con $t \in \{1, \cdots, n, \cdots, n+m\}$, è definita da:

$$\begin{aligned} p_t(2i) &= \sin\left(t/10000^{2i/d^p}\right) \\ p_t(2i-1) &= \cos\left(t/10000^{2i/d^p}\right) \end{aligned} \tag{19}$$

dove d^p è la dimensione di embedding, i è l'indice e $1 \leqslant i \leqslant [d^p/2]$. Ogni dimensione dell'embedding di posizione è una sinusoide.

Le lunghezze d'onda formano una progressione geometrica da 2π a $10000 \cdot 2\pi$. Dato un offset k, si può rappresentare p_{t+k} come funzione lineare di p_t. Pertanto, questa soluzione consente al modello di imparare a riconoscere le posizioni relative e prevedere pose dall'aspetto naturale a diversi intervalli di tempo. La rete diviene inoltre capace di estrapolare sequenze di lunghezza variabile durante l'allenamento.

In ingresso al Decoder vengono fornite le stesse tipologie di input dell'Encoder: per prevedere la posa all'istante $t+1$ servono quindi le informazioni relative al passo precedete, ovvero la posa predetta x_t, la sua velocità v_t e la codifica di posizione p_t.

La formulazione matematica per l'aggiornamento dell'*hidden state* h_t, valida sia per l'encoder che per il decoder, per un certo istante t è la seguente:

$$\begin{aligned} z_t &= \sigma\left(U_x^z x_t + U_v^z v_t + U_p^z p_t + W^z h_{t-1}\right) \\ r_t &= \sigma\left(U_x^r x_t + U_v^r v_t + U_p^r p_t + W^r h_{t-1}\right) \\ \hat{h}_t &= \tanh\left(U_x^h x_t + U_v^h v_t + U_p^h p_t + W^h\left(r_t \circ h_{t-1}\right)\right) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \end{aligned} \tag{20}$$

dove r_t è il *recurrent gate*, z_t l'*update gate* e U, W sono rispettivamente le matrici delle variabili e dei pesi.

Il Decoder prevede in prima istanza la velocità per l'istante $t+1$ e, tramite la posa x_t, ricava la posa successiva x_{t+1}:

$$\begin{aligned} v_t &= Wh_t + b \\ x_{t+1} &= x_t + v_t \end{aligned} \tag{21}$$

W e b rappresentano i pesi e il bias del layer che effettua questa operazione, posizionato dopo la GRU.

Per concludere, il risultato finale viene ottenuto convertendo la notazione da asse-angolo a quaternioni attraverso un apposito layer di elaborazione.

2.3.3 *Trasformazione in Quaternioni*

Per convertire gli angoli nella notazione dei quaternioni, PVRED utilizza un layer denominato QT, visibile in Figura 5. Dato un vettore di tre dimensioni in rappresentazione asse-angolo e, il layer QT lo trasforma in un quaternione q nel seguente modo:

$$q(i) = \begin{cases} \cos(0.5\|e\|_2) & i = 1 \\ \frac{\sin(0.5\|e\|_2)}{\|e\|_2} \cdot e(i-1) & i \geqslant 2 \end{cases} \tag{22}$$

dove i rappresenta l'elemento i-esimo di q.

2.3.4 *Allenamento della rete*

L'addestramento del modello avviene considerando una *loss function* nello spazio dei quaternioni unitari. Naturalmente, l'obiettivo consiste nel ridurre al minimo la discrepanza tra le posizioni previste e quelle effettive, allo stesso tempo assicurando che la lunghezza del quaternione predetto rimanga unitaria. La funzione utilizzata è:

$$L_{QT} = \frac{1}{m} \sum_{j=1}^{m} \left\| g\left(\hat{x}_j\right) - g\left(x_j\right) \right\| \tag{23}$$

dove g indica la trasformazione in quaternioni, $\hat{x}_j$ indica la posa predetta e x_j è la posa reale.

La rete è stata allenata dai suoi autori sui dataset Human3.6M e CMU (vedi 1.3.1) utilizzando come input i giunti in notazione asse-angolo. L'unica ulteriore elaborazione dei dati originali presenti nei dataset è la riduzione del framerate a 25 fps.

2.3.5 *Modifiche implementate*

Dei dataset utilizzati nell'analisi effettuata in questa tesi (H3.6M [8] e AMASS [14]), solo H3.6M è stato utilizzato dagli autori di PVRED. Si è reso quindi necessario adattare l'implementazione originale per renderla compatibile con la diversa rappresentazione dello scheletro di AMASS. Sono state inoltre aggiunte le metriche mancanti tra quelle descritte in sezione 3.1 per comparare i risultati ottenuti da PVRED con quelli delle altre reti.

3

METRICHE

Al fine di valutare quantitativamente i risultati degli esperimenti descritti nel Capitolo 4, vengono di seguito introdotte e descritte le metriche per valutare le prestazioni delle architetture considerate. Viene inoltre introdotto il modello *Zero-Velocity*, una base di riferimento molto semplice, adoperata per giudicare se l'impiego di reti neurali complesse conduca a risultati predittivi superiori a quelli ottenibili con tale modello.

3.1 DESCRIZIONE DELLE METRICHE

Per misurare la qualità della previsione del movimento, è necessario confrontare l'intera sequenza prevista con quella target, ovvero la sequenza reale. Data la complessità della struttura del corpo umano, le metriche dovrebbero tenere conto di tutta la sequenza disponibile e di tutti gli angoli o coordinate che descrivono la posa. La formulazione si basa sulla descrizione matematica della rappresentazione del movimento umano introdotta in 1.3.2. Per ciascuna metrica viene fornita la definizione per un dato frame t e per un dato campione di test X_{test}.

3.1.1 *Angoli di Eulero*

La metrica attualmente utilizzata come standard nel campo della previsione del movimento umano è l'errore degli Angoli di Eulero. Si basa sull'impiegare una specifica sequenza di Eulero (es. ZXY) per descrivere gli angoli dei giunti predetti e reali e calcolare la distanza Euclidea. Nonostante alcune limitazioni (vedi 1.3.2), rimane un metodo valido per confrontare le prestazioni dei vari modelli esistenti e funge da riferimento significativo per valutare l'accuratezza delle previsioni degli angoli articolari.

La metrica degli Angoli di Eulero, indicata con $L_{eul}(t)$, utilizzata per valutare l'errore all'istante di tempo t si calcola come segue:

$$L_{eul}(t) = \frac{1}{|X_{test}|} \sum_{x_t \in X_{test}} \sum_{j=1}^{J} \left\| \hat{\alpha}_j - \alpha_j \right\|_2 \tag{24}$$

dove $\hat{\alpha}_j$ e α_j rappresentano la stima e il valore reale all'istante t della rotazione della j-esima articolazione espressa nella notazione degli Angoli di Eulero. La metrica degli Angoli di Eulero valuta quindi la

distanza Euclidea media tra gli Angoli di Eulero previsti e quelli reali su tutti i campioni nel set di test.

3.1.2 *Differenza tra gli Angoli Articolari*

Introdotta in [1] per attenuare possibili errori associati alla metrica dell'Angolo di Eulero, la Differenza tra gli Angoli Articolari è una metrica alternativa basata sugli angoli, che quantifica l'angolo di rotazione necessario per allineare la posizione prevista del giunto con la posizione obiettivo. Conosciuta anche come metrica geodesica, a differenza della metrica degli Angoli di Eulero, è indipendente dalla specifica parametrizzazione delle rotazioni. La Differenza tra gli Angoli Articolari si calcola come segue:

$$L_{\text{angle}}(t) = \frac{1}{|X_{\text{test}}|} \sum_{x_t \in X_{\text{test}}} \frac{1}{J} \sum_{j=1}^{J} \left\| \log\left(\tilde{R}_j\right) \right\|_2 \tag{25}$$

dove $\tilde{R}_j$ è la matrice di rotazione necessaria per allineare la predizione del giunto j con la sua orientazione reale, presupponendo che i giunti precedenti lungo la catena cinematica siano già allineati correttamente. Analogamente a quanto visto per gli Angoli di Eulero, la metrica per la sequenza complessiva si ottiene applicando la media su tutti i frame della previsione.

3.1.3 *Errore di Posizione*

Al fine di valutare l'accuratezza delle previsioni relative alle posizioni assolute delle articolazioni, si fa ricorso all'errore di posizione, indicato con $L_{\text{pos}}(t)$.

La metrica dell'errore di posizione consente un punto di vista differente sull'errore complessivo e può essere più adatta o meno rispetto alle precedenti metriche a seconda del campo di applicazione. Infatti, se l'obiettivo della previsione è conoscere la posizione delle estremità del corpo, come le mani, al fine di collaborare con un robot, allora questa è la metrica di maggiore interesse. Tuttavia, se si intende prevedere l'intera posa della persona per valutare gli indici ergonomici o per riprodurla in un ambiente virtuale, le metriche da considerare maggiormente saranno quelle relative agli angoli, poiché, in questi esempi, risulta più importante che l'intera posa sia verosimile piuttosto che le posizioni assolute siano molto precise.

In questa metrica viene messo in risalto il fatto che gli errori nelle articolazioni iniziali dell'albero cinematico (come la colonna vertebrale e le spalle) influenzano gli errori delle altre articolazioni. Le articolazioni iniziali hanno infatti un notevole impatto sulle posizioni globali delle articolazioni successive come ad esempio polsi o caviglie, generando un effetto amplificato sulla precisione complessiva

della previsione. Di conseguenza, valutare la metrica dell'errore di posizione consente di avere un quadro complessivo dell'errore, consentendo di considerare l'effetto cumulativo degli errori attraverso varie articolazioni anziché concentrarsi unicamente sugli errori relativi alle singole articolazioni.

Questa metrica si basa sul calcolo della cinematica diretta a partire dagli angoli delle articolazioni reali α e quelle previste $\hat{\alpha}$ al tempo t. Questo processo fornisce le posizioni 3D delle articolazioni p e $\hat{p}$. Successivamente, viene calcolata la distanza euclidea tra ciascuna coppia di posizioni articolari corrispondenti. La formulazione matematica è la seguente:

$$L_{pos}(t) = \frac{1}{|X_{test}|} \sum_{x_t \in X_{test}} \frac{1}{J} \sum_{j=1}^{J} \left\| \hat{p}_j - p_j \right\|_2 \tag{26}$$

dove $|X_{test}|$ rappresenta il campione di test, x_t denota un singolo frame al tempo t, J indica il numero totale di giunti, p_j e $\hat{p}_j$ fanno riferimento rispettivamente alle posizioni reali e predette dei giunti. Per garantire la coerenza dei risultati, nel calcolo della cinematica diretta le lunghezze delle ossa dello scheletro vengono normalizzate, con il femore destro che funge da riferimento di lunghezza unitaria come suggerito in [1].

Come visto nella sezione 2.2.4, il calcolo della cinematica diretta comporta una perdita d'informazione. Questo è un ulteriore aspetto da considerare se si utilizza questa metrica, infatti, la coincidenza delle posizioni dei giunti non implica necessariamente la corrispondenza delle loro rotazioni.

3.1.4 *Valutazione Qualitativa*

Un altro approccio che può essere utilizzato per valutare la qualità delle sequenze previste è una valutazione qualitativa basata sulla visualizzazione grafica. Sebbene non si tratti di una metrica oggettiva, risulta spesso molto efficace poichè è di immediata interpretazione e consente di evidenziare errori banali che sarebbero tuttavia quasi impossibili da rilevare utilizzando solo metriche quantitative. Ad esempio, si consideri un modello le cui previsioni sono ottime per la parte superiore del corpo ma completamente sbagliate per la parte inferiore. In questo caso, le metriche sopra descritte non sono in grado di evidenziare il problema perché considerano tutte le articolazioni dello scheletro. Invece, visualizzando diverse sequenze previste, questo problema verrebbe facilmente rilevato.

La visualizzazione delle sequenze si dimostra un utile strumento specialmente se si sovrappone la previsione con la sequenza reale. In questo modo, diventa immediato riconoscere quali giunti presentano l'errore maggiore. Naturalmente, considerazioni simili potreb-

bero essere fatte calcolando i parametri in modo indipendente per ciascun giunto e confrontandoli; tuttavia, questo approccio renderebbe la valutazione lunga e complessa, una visualizzazione grafica è quindi preferibile in situazioni di questo tipo.

3.2 MODELLO BASE: ZERO-VELOCITY

Per valutare la qualità della previsione dei vari modelli, vengono utilizzate le metriche sopra analizzate, tuttavia, questi parametri da soli potrebbero non fornire una valutazione completa delle prestazioni. Pertanto, è fondamentale confrontare i risultati ottenuti dai vari modelli con quelle di una base di riferimento adeguata. Una possibile base per il confronto è data dal modello *Zero-Velocity*, anche se non è l'unica: esistono varie possibilità, come la media mobile usata in [17].

Il modello Zero-Velocity può essere definito come una ripetizione continua dell'ultimo frame osservato. Ciò significa che ogni frame previsto sarà una replica esatta dell'ultimo frame, quindi gli angoli articolari previsti non varieranno per tutta la predizione. Di conseguenza, il risultato corrisponde a una previsione interamente statica, ovvero con velocità pari a zero, da cui deriva la denominazione del modello.

Nella sua semplicità, il modello Zero-Velocity risulta essere un metro di paragone ideale. Non deve infatti essere sottovalutato, poiché in molteplici situazioni si è dimostrato una sfida superarlo, soprattutto considerando la metrica degli Angoli di Eulero, come evidenziato in [1]. Prevedere che una persona si muoverà è infatti generalmente più rischioso rispetto allo stabilire che rimarrà ferma. Questo succede perché, se il movimento previsto è nella direzione opposta rispetto al movimento reale, l'errore sarà il doppio rispetto a quello di una previsione stazionaria. Dati i molti fattori che possono influenzare il movimento di una persona e la somiglianza tra varie pose del corpo che portano a movimenti differenti, non è raro fare una previsione che vada nella direzione opposta a quella reale. Pertanto, il confronto con il modello a velocità zero fornisce preziose informazioni sulle prestazioni dei modelli e sulla loro capacità di effettuare una previsione degli angoli migliore del modello base.

3.3 METRICHE USATE IN LETTERATURA

La scelta delle metriche da utilizzare per valutare le prestazioni dei diversi modelli è un aspetto fondamentale. Infatti, se i risultati sono complessi e con diverse sfumature, le conclusioni che si traggono da una particolare metrica possono non essere in linea con quanto si ricava se ne viene considerata una differente. Per questo motivo, risulta importante esaminare quali metriche sono state utilizzate dai ricercatori per valutare l'efficacia e l'accuratezza dei metodi proposti. In

particolare, in Tabella 1 si riportano le metriche utilizzate nei lavori presentati in 1.3.3.

Tabella 1: Metriche utilizzate dalle diverse architetture per la previsione del movimento umano.

	Angoli di Eulero	Differenza tra gli Angoli Articolari	Errore di Posizione
DMGNN	✓		
RNN-MTP			✓
HRI	✓		✓
LPJP	✓		✓
PVRED	✓		
STT	✓	✓	✓
DMMGAN			✓
BiTGAN			✓

Tra le reti in elenco, si osserva che solo STT utilizza tutte e 3 le metriche considerate in questa tesi, mentre più della metà delle reti elencate ne adopera solamente una.

4

ESPERIMENTI

Al fine di identificare quale architettura sia più adatta ad essere utilizzata per prevedere il movimento umano nell'ambito della robotica collaborativa, sono stati condotti numerosi test sia utilizzando dataset preesistenti che con un nuovo dataset raccolto appositamente per questa tesi. In questo Capitolo viene descritta l'analisi preliminare effettuata su AMASS e Human3.6M, il setup sperimentale progettato e realizzato per la raccolta del dataset, la tipologia di azioni registrate e l'elaborazione dei dati necessaria per convertirli nel formato SMPL da fornire in input alle varie reti neurali.

4.1 ANALISI PRELIMINARE CON H3.6M E AMASS

Un primo confronto tra le architetture selezionate è stato effettuato utilizzando dei dataset di movimenti umani generici. In questo modo è infatti possibile identificare se le prestazioni di una specifica architettura siano nettamente migliori rispetto alle altre o se approssimativamente i risultati si equivalgano. I dataset scelti per questa comparazione sono H3.6M e AMASS, il primo per via del suo già ampio utilizzo nell'ambito della previsione del movimento, il secondo grazie alla sua vasta quantità e diversità di dati presenti.

Il dataset H3.6M è stato usato nella sua interezza, utilizzando come set di *training*, *test* e *validation* i dati dei seguenti soggetti: $S_{\text{training}} = [S1, S6, S7, S8, S9]$, $S_{\text{test}} = [S11]$, $S_{\text{validation}} = [S5]$, ovvero seguendo le convenzioni genericamente adottate da quando si utilizza questo dataset (si veda ad esempio [10, 16, 17, 29]).

Dalla collezione di dataset AMASS sono stati selezionati quelli che soddisfano la maggior parte dei seguenti criteri:

- *Descrizione delle azioni in BABEL [24]:* la presenza di una descrizione testuale per ogni micro-azione in una registrazione consente di progettare delle reti neurali che sfruttino la connessione tra le pose e la relativa etichetta. Riconoscendo quindi la categoria dell'azione, diventa possibile effettuare una previsione nettamente più accurata su come si muoverà la persona.

- *Presenza di almeno sei soggetti differenti:* la presenza di diverse persone che eseguono lo stesso tipo di azioni consente alle reti di generalizzare meglio l'azione imparata grazie alla variabilità nello svolgimento tra un soggetto e l'altro. Il valore di almeno sei o sette è scelto basandosi sulla quantità di soggetti in H3.6M.

- *Lunghezza complessiva delle registrazioni elevata:* al fine di avere una quantità sufficiente di dati vengono selezionati i dataset in cui la durata complessiva di tutte le registrazioni sia maggiore di circa quindici minuti.
- *Utilizzo in pubblicazioni che sfruttano AMASS:* nonostante non sia fondamentale, si è cercato di utilizzare dataset che siano già stati scelti in architetture nell'ambito della predizione del movimento umano, coma ad esempio HRI [16] o STT [2].
- *Esecuzione di azioni generiche:* le azioni eseguite nel dataset devono essere varie e non troppo specifiche o particolari. Ad esempio, non sono adatti dataset focalizzati su una singola azione come l'afferrare oggetti o dataset in cui le persone si mettono volutamente in pose non naturali.

In Tabella 2 sono elencati i dataset utilizzati e le relative informazioni su lunghezza e numero di Mocap (abbreviazione di *Mo*tion *Cap*ture, con cui si intende una singola registrazione di una persona).

Tabella 2: Elenco dei dataset di AMASS usati per l'analisi. I dati sono ottenuti considerando Mocap ricampionati ad un framerate di 60fps.

Dataset	Soggetti	Mocap	Frame	Durata [Minuti]	Frame medi per Mocap
ACCAD [26]	20	181	85 671	23,80	473,32
BMLmovi [6]	89	1 611	578 965	160,82	359,38
BMLrub [6]	111	2 523	1 817 725	504,92	720,46
CMU [4]	96	1 831	1 956 275	543,41	1 068,42
EKUT [15]	4	314	105 433	29,29	335,77
EyesJapanDataset [13]	12	750	1 309 016	363,62	1 745,35
HDM05 [19]	4	204	518 928	144,15	2 543,76
MoSh [12]	19	77	59 488	16,52	772,57
SFU [23]	7	42	54 513	15,14	1 297,93
TotalCapture [25]	5	37	147 951	41,10	3 998,68
Transitions [22]	1	110	54 319	15,09	493,81
Totale	368	7 680	6 688 284	1 857,86	870,87

Sono state inoltre scartate le sequenze più brevi di tre secondi, valore scelto in modo tale da utilizzare i primi due secondi come input ai modelli e il terzo secondo come confronto per verificare la qualità della previsione. Ad esempio, se i dati sono a 25fps, si otterrano 50 frame per l'input e 25 per l'output, quindi tutte le sequenze più brevi di 75 frame non vengono considerate. Le sequenze valide vengono poi divise in modo casuale tra *training*, *test* e *validation* secondo le percentuali 90%, 5%, 5%. Riservare il 5% delle sequenze per le fasi di *test*

e *validation* corrisponde ad utilizzare un totale di circa 330 000 frame cioè più di tre volte volte la dimensione del set di *validation* di H3.6M (circa 100 000 frame a 50fps, cioè il 20% dei frame totali).

Tutti i modelli sono stati allenati e testati su un Cluster di calcolo le cui caratteristiche sono in Tabella 3.

Tabella 3: Specifiche del Cluster di calcolo utilizzato per l'elaborazione dei dati.

GPU	3x NVIDIA Tesla V100 S
GPU (memoria)	3x 32 GB
CPU	Intel Xeon CPU E5-2609 V3
RAM	64 GB
Storage	30 TB

4.2 SETUP SPERIMENTALE PER LA RACCOLTA DEL DATASET

Il dataset raccolto per questa tesi è stato realizzato registrando diverse azioni tipiche di una stazione di assemblaggio con numerose fasi di collaborazione con un robot. Questo dataset è stato registrato nel C-Square Lab e viene identificato con la sigla CS_CORO (da *CO*llaborative *RO*bot). Il setup sperimentale è composto da quattro telecamere Azure Kinect e un'insieme di sensori inerziali indossati dall'operatore. Di seguito vengono descritti i sensori utilizzati, le attrezzature predisposte e il robot impiegato durante la registrazione del dataset.

- *XSens IMU:* le *inertial measurement unit* sono dei dispositivi wireless di dimensioni contenute che, applicati al corpo tramite adesivo o velcro, consentono di fornire informazioni di posizione e orientamento con una elevata frequenza. Generalmente le IMU si compongono di tre giroscopi, di tre accelerometri e di tre magnetometri che consentono così di misurare i movimenti angolari e quelli lineari in ogni direzione ricavando sufficienti informazioni per descrivere un corpo rigido nello spazio. Disponendo le IMU in vari punti chiave del corpo umano diventa quindi possibile ricavare gli angoli formati da ogni giunto e stabilire così la posa della persona con grande precisione. Nella raccolta del dataset vengono utilizzate 17 IMU *MTw Awinda* di *XSens*, una di queste visibile in Figura 6.

- *XSens Metagloves:* i guanti *XSens Metagloves by Manus* sono dispositivi indossabili progettati per catturare i movimenti delle mani e delle dita con elevata precisione. Ogni guanto è dotato di sensori di movimento su ogni dito della mano, giroscopi e

accelerometri registrano i cambiamenti nella posizione e nell'orientamento con precisione nell'ordine dei millimetri. Inoltre, il peso di soli 70g e la connettività wireless consentono un utilizzo confortevole per lunghi periodi. Nella Figura 7 è mostrato uno dei guanti utilizzati da tutti i partecipanti durante la registrazione del dataset, da cui sono state raccolte informazioni dettagliate riguardo ai movimenti delle mani nel contesto di collaborazione uomo-robot.

- *Azure Kinect:* le telecamere *Azure Kinect* (Figura 8) sono dei dispositivi sviluppati da *Microsoft* per applicazioni di visione artificiale e rilevamento di movimento. Dispongono di un sensore di profondità, una videocamera a colori, un accelerometro, un giroscopio, un array di microfoni e un sistema di sincronizzazione con altri dispositivi. Nella raccolta del dataset sono state utilizzate 4 Kinect disposte intorno alla zona di lavoro in modo da registrare la scena da diversi punti di vista. La possibilità di acquisire le informazioni video da molteplici angolazioni ha consentito di minimizzare le occlusioni che avrebbero ridotto la quantità di informazioni in determinate azioni. Tramite l'uso di queste telecamere è inoltre possibile stimare la posa di una persona.

- *Zona di lavoro:* la zona di lavoro è stata realizzata secondo gli standard ergonomici utilizzati nella progettazione delle stazioni di assemblaggio industriale. Il tavolo principale ha un'altezza di 95cm, fattore di fondamentale importanza per effettuare lavorazioni in posizione eretta (per maggiori dettagli a questo riguardo si veda [7]). Sono inoltre presenti dei contenitori per il prelievo ad altezza spalle che contengono le componenti di uso comune mentre gli utensili sono posizionati a lato dell'operatore in modo che la presa risulti confortevole. In aggiunta, vi sono degli scaffali dietro all'operatore dove riporre o raccogliere oggetti voluminosi. Infine, il robot si trova dalla parte opposta del tavolo rispetto all'operatore, in questo modo risulta possibile massimizzare l'area di lavoro condivisa senza compromettere la sicurezza del sistema. In Figura 9 si può osservare la zona di lavoro realizzata.

- *Robot:* per interagire con l'operatore nelle varie azioni registrate nel dataset è stato utilizzato il robot collaborativo *Franka Emika Panda* (Figura 10), un braccio robotico a sette gradi di libertà con capacità di carico fino a 3Kg, estensione massima di 850mm, ripetibilità di $0,1$mm e peso di circa 18Kg.

Figura 6: Sensore inerziale MTw Awinda di XSens per la raccolta dei dati sulla posa della persona. Tra le informazioni raccolte sono compresi gli angoli articolari, le posizioni assolute di diversi punti del corpo e il centro di massa.

Figura 7: Guanti XSens Metagloves by Manus per la registrazione accurata delle posizioni e dei movimenti delle dita.

Figura 8: Una delle telecamere Azure Kinect utilizzate per registrare le azioni nel dataset. Ognuna dispone di un sensore di profondità, una videocamera a colori, un accelerometro, un giroscopio e un array di microfoni.

Figura 9: Foto dell'area operativa allestita per la raccolta del dataset.

Figura 10: Robot *Franka Emika Panda* utilizzato per la collaborazione con l'operatore nella raccolta del dataset.

4.3 AZIONI REGISTRATE

Per la creazione di CS_CORO, sono stati delineati cinque compiti distinti da affidare alla persona in fase di registrazione. Ognuno di questi coinvolge l'utilizzo di uno o più strumenti tra cui un cacciavite, un avvitatore, un trapano, un martello e un saldatore. Le operazioni effettuate dal robot sono di diversa natura in base alla tipologia di attività eseguita dall'operatore: in alcuni casi effettua il trasporto del pezzo su cui devono essere effettuate le operazioni e lo ruota a seconda delle necessità, in altre situazioni è incaricato di portare le viti o i bulloni adatti per il compito che si sta eseguendo e, quando non direttamente impiegato nell'incarico dell'operatore, effettua operazioni necessarie per velocizzare l'esecuzione delle attività successive.

L'obiettivo principale che ha guidato la progettazione delle diverse fasi di ogni attività è stato rendere tali fasi plausibili ed ergonomiche, considerandole in un contesto industriale di collaborazione uomo-robot. Inoltre, la libertà lasciata ad ogni soggetto nello scegliere l'ordine con cui prendere oggetto e strumento o dove avvitare le viti o inserire i chiodi consente di ottenere un certo grado di variabilità tra le azioni registrate, un fattore importante per valutare le prestazioni generali di una rete neurale.

Complessivamente sono stati raccolte le azioni di 5 soggetti per un totale di circa 20 minuti di registrazione.

Di seguito vengono presentate tutte le fasi che compongono ognuna delle attività registrate, indicate tramite gli utensili che vengono utilizzati. Per semplificare la lettura vengono abbreviati *Robot* (R) e *Operatore* (Op). Va notato inoltre che le varie attività sono indipendenti tra loro e possono essere state registrate in un ordine differente da quello presentato.

A) *Cacciavite e Saldatore*

1. Op prende il cacciavite dal tavolo e due viti,
2. R raccoglie un componente e lo avvicina ad Op con un'inclinazione che renda ergonomiche le operazioni successive,
3. Op avvita le viti sul componente,
4. R ruota il componente mostrandone il retro,
5. Op nel frattempo ripone il cacciavite e prende il saldatore,
6. Op salda sul componente,
7. R si allontana e deposita il componente finito,
8. Op nel frattempo ripone il saldatore.

B) *Avvitatore*

1. Op prende un componente dai contenitori di prelievo,

2. R riconosce l'azione e porta ad Op un contenitore con i bulloni da utilizzare per questa specifica fase,
3. Op prende l'avvitatore e avvita due bulloni sul componente,
4. R riporta il contenitore con i bulloni nella sua zona di lavoro,
5. Op ripone l'avvitatore e poi impila il componente finito sopra ad altri componenti uguali,
6. Op porta la pila di componenti finiti sullo scaffale dietro la postazione di lavoro.

C) *Nastro Adesivo*

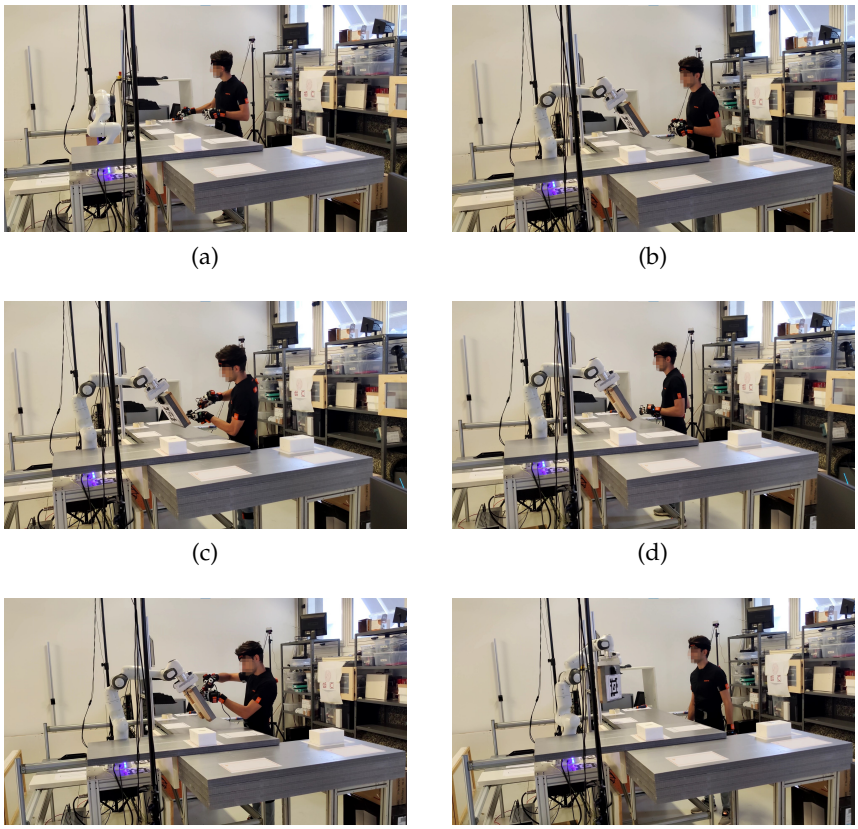
1. Op prende uno scatolone dagli scaffali,
2. R riconosce l'azione e porta ad Op un componente finito,
3. Op prende il componente da R che ritorna nella sua area di lavoro,
4. Op inserisce il componente nello scatolone e lo sigilla con il nastro adesivo,
5. Op sposta lo scatolone in una zona apposita dove ci sono altri scatoloni chiusi.

D) *Martello*

1. Op prende un componente dalla zona di prelievo del tavolo,
2. R riconosce di non essere coinvolto in questa attività e quindi effettua altre operazioni in una zona distante da Op,
3. Op prende il martello e due chiodi e li inserisce nel componente,
4. Op ripone il martello e il componente nelle zone designate.

E) *Trapano*

1. Op prende un componente dal tavolo e lo porta nella zona del bancone adibita all'uso del trapano,
2. R porta un contenitore con la punta del trapano adeguata per la lavorazione da effettuare,
3. Op raccoglie il trapano e vi inserisce la punta che gli ha portato R,
4. Op realizza due fori sul componente,
5. Op rimuove la punta dal trapano e la restituisce ad R che riporta indietro il contenitore.



(a) (b) (c) (d) (e) (f)

Figura 11: Sequenza di azioni effettuate da uno dei soggetti registrati durante l'esecuzione dell'attività A.

In Figura 11 viene riportata la sequenza di azioni che sono state eseguite da parte di uno dei soggetti durante la registrazione del dataset. Nello specifico vengono mostrate le diverse fasi dell'attività A.

4.4 ELABORAZIONE DATI

La trasmissione dei dati misurati dai sensori inerziali, le riprese delle telecamere e le altre informazioni, come ad esempio il posizionamento relativo delle telecamere nello spazio, avviene tramite ROS.

ROS è un middleware che consente lo scambio di informazioni nei sistemi distribuiti con un alto livello di astrazione per l'utilizzatore, permette infatti ad un programma di pubblicare dei messaggi su un certo argomento chiamato *topic* e questi messaggi sono personalizzabili in base alle necessità e preferenze dell'utente. Uno dei vantaggi dell'utilizzo di ROS è la possibilità di registrare su un file chiamato *rosbag* i topic che vengono pubblicati dai vari sensori, ad ogni messaggio è inoltre collegato il relativo *timestamp* (l'istante temporale) di quando è stato generato.

Per ogni attività di un soggetto è stata quindi registrata una rosbag

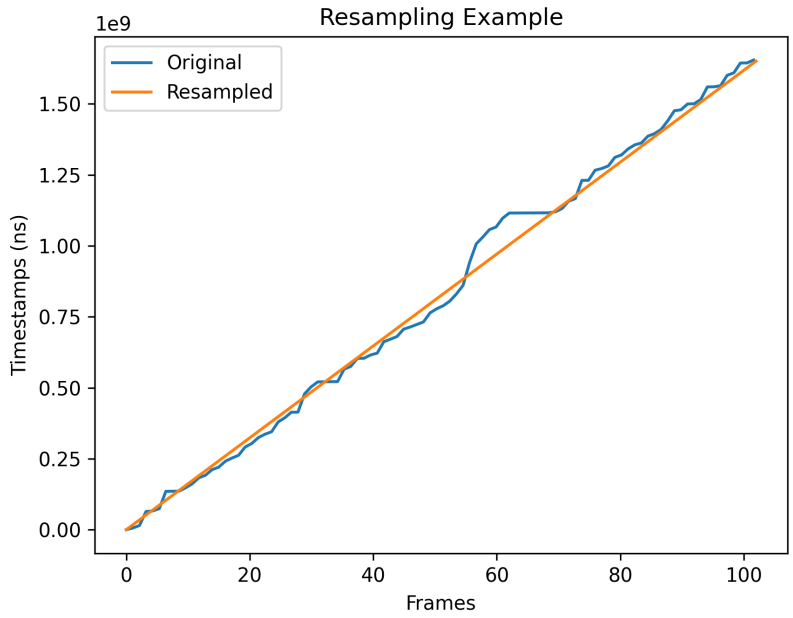


Figura 12: Esempio del ricampionamento dei dati sui primi 100 frame di una registrazione. I timestamp sono in nanosecondi e vengono traslati in modo che il primo frame abbia timestamp $= 0$. Si può osservare che i timestamp originali non sono ad intervalli costanti mentre nel segnale ricampionato lo sono.

contenente i seguenti dati che verranno in seguito elaborati al fine di ottenere una descrizione della posa della persona compatibile con quanto presente nel dataset AMASS:

- Dati delle IMU:
  - Posizione assoluta di diversi punti del corpo (i *markers*).
  - Centro di massa della persona.
  - Angoli di tutte le articolazioni del corpo.
- Dati dei guanti:
  - Posizione assoluta delle dieci dita.
  - Orientazione di ogni falange delle mani.
- Matrici di Trasformazione (TF) relative alla posizione nello spazio e orientazione delle telecamere e del cobot, rispetto ad un sistema di riferimento globale.
- Video RGB delle quattro telecamere Kinect.

Le informazioni più importanti che vengono raccolte sono quelle relative agli angoli misurati dall'insieme dei sensori inerziali. Tra i 28 giunti XSens disponibili, alcuni vengono esclusi in quanto contengono dati sintetici dedicati alle analisi ergonomiche e non rilevanti

nella valutazione delle reti neurali sotto esame. Gli altri giunti vengono invece mappati secondo la struttura del modello SMPL (vedi Figura 1).

Un altro elemento di notevole importanza è il primo dei markers poiché contiene la posizione assoluta del bacino della persona. Questo valore viene utilizzato per identificare la posizione e quindi il movimento del soggetto durante la registrazione rispetto al sistema di riferimento globale. Nonostante non venga attualmente utilizzata nelle reti sotto esame, è comunque un'informazione fondamentale per future evoluzioni delle architetture di previsione sotto esame, viene quindi salvata assieme ai dati dei giunti.

Un'ultima operazione da effettuare nell'elaborazione dei dati consiste nel ricampionarli in modo tale che tra un frame e il successivo ci sia sempre un intervallo di tempo costante (ad esempio $40\mathrm{ms}$ se il framerate è $25\mathrm{fps}$). Come si può vedere in Figura 12 i dati non vengono acquisiti ad intervalli costanti, tuttavia, grazie al timestamp di ogni messaggio, è possibile trattare ogni angolo dei giunti come un segnale digitale e ricampionarlo ad istanti temporali regolari.

5 RISULTATI

In questo capitolo vengono presentati i risultati degli esperimenti descritti nel Capitolo 4. Come riportato nei capitoli precedenti, le analisi sono state effettuate utilizzando le architetture DMGNN, HRI, PVRED e il modello base Zero-Velocity. Le metriche utilizzate sono l'errore degli Angoli di Eulero, la Differenza degli Angoli Articolari e l'Errore di Posizione. Le valutazioni sono state effettuate su H3.6M, AMASS e su CS_CORO, il dataset raccolto appositamente per valutare in contesti collaborativi le prestazioni delle diverse architetture.

Tutti i risultati presentati sono ad una frequenza di 25fps e su di un orizzonte temporale di 1000ms, ottenendo quindi una previsione di 25 frame in avanti.

L'Errore di Posizione viene riportato in scala percentuale, dove il 100% corrisponde all'errore massimo ottenuto tra le varie reti. Ciò è necessario al fine di consentire il confronto tra i diversi dataset, poiché nel calcolo delle posizioni assolute dei giunti, le dimensioni dello scheletro influenzano il valore assoluto dei risultati.

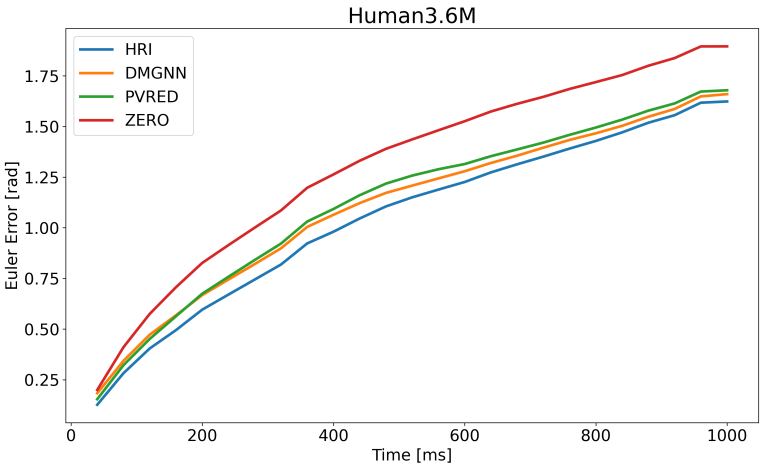
5.1 RISULTATI SU H3.6M

In Figura 13 vengono illustrati i risultati delle tre metriche considerate per valutare le predizioni delle varie reti. In un dataset relativamente piccolo e con una quantità limitata di azioni come H3.6M, tutte le architetture considerate riescono ad effettuare delle previsioni efficaci, infatti i risultati che ottengono sono migliori del modello Zero-Velocity per ogni metrica. Confrontando i risultati di tutte le metriche si può inoltre osservare che le previsioni a breve termine di DMGNN e PVRED sono quasi equivalenti, con DMGNN che sul lungo termine riesce a generare delle predizioni leggermente migliori.

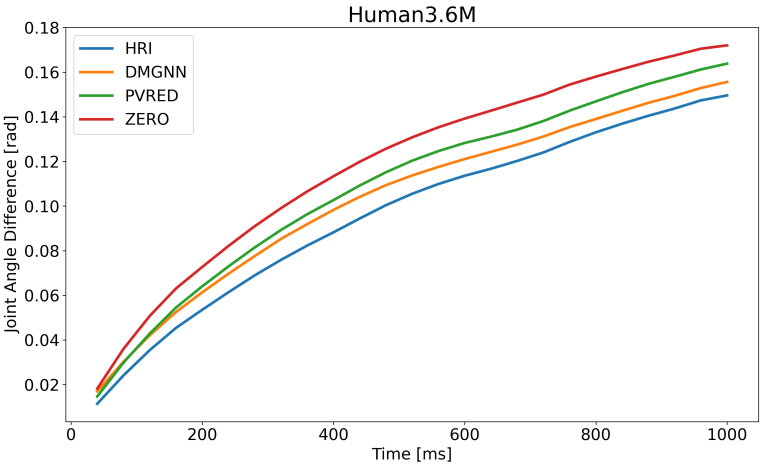
Inoltre, da questa prima analisi basata sul dataset maggiormente utilizzato come benchmark, si evince che HRI risulta essere la rete con le prestazioni migliori. Essa ottiene degli errori inferiori per ogni frame di previsione in ognuna delle metriche valutate.

5.2 RISULTATI SU AMASS

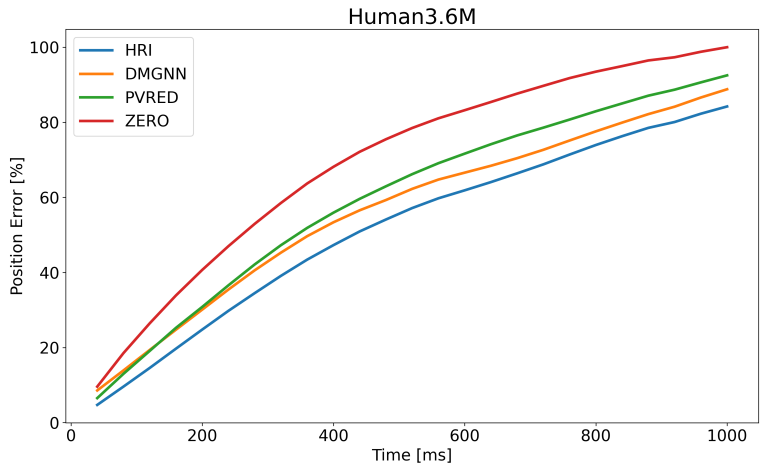
I risultati ottenuti su di un dataset ampio ed eterogeneo come AMASS, visibili in Figura 14, sono meno definiti e univoci rispetto a quanto visto sopra per H3.6M. Infatti, non c'è una rete le cui previsioni siano sempre le migliori per ognuna delle metriche considerate. Da un



(a) Errore degli Angoli di Eulero su H3.6M.

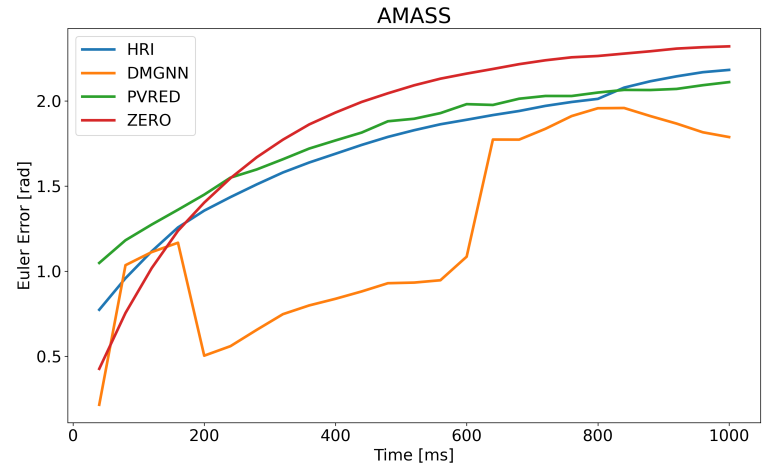


(b) Differenza degli Angoli Articolari su H3.6M.

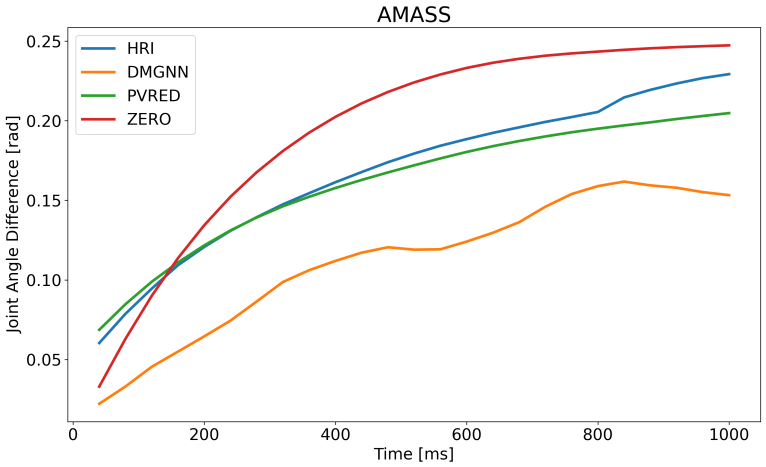


(c) Errore di Posizione su H3.6M.

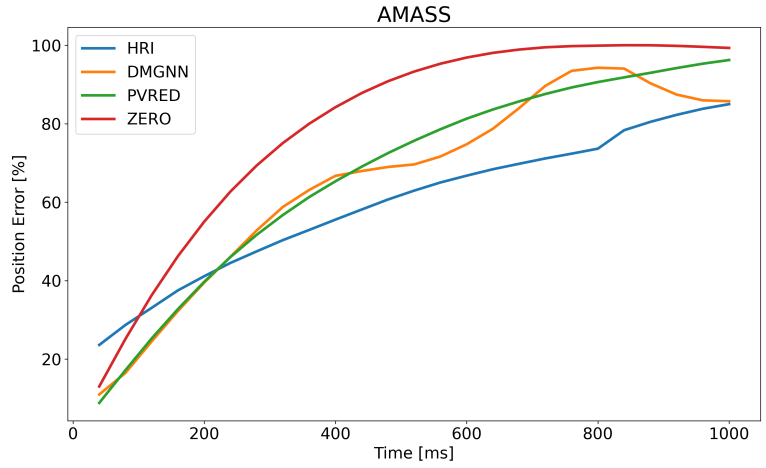
Figura 13: Risultati di DMGNN, HRI, PVRED e Zero-Velocity su H3.6M.



(a) Errore degli Angoli di Eulero su AMASS.



(b) Differenza degli Angoli Articolari su AMASS.



(c) Errore di Posizione su AMASS.

Figura 14: Risultati di DMGNN, HRI, PVRED e Zero-Velocity su AMASS.

punto di vista generale si può inoltre osservare che nei primi frame di previsione il modello Zero-Velocity ottiene risultati migliori di alcune delle reti.

Analizzando nello specifico la metrica degli Angoli di Eulero in Figura 14a, si osserva un andamento particolare per DMGNN: nel brevissimo termine (meno di 200ms) la rete fatica molto nelle previsioni, salvo poi ottenere degli ottimi risultati per l'intervallo $[200, 600]$ ms. Gli andamenti di HRI e PVRED sono più lineari, tuttavia, anche in questi casi i primi 200ms sono critici perchè l'errore è superiore a quello dello Zero-Velocity. Inoltre, circa ad 800ms i risultati di HRI peggiorano leggeremente portandola a previsioni peggiori di PVRED per l'intervallo di tempo successivo.

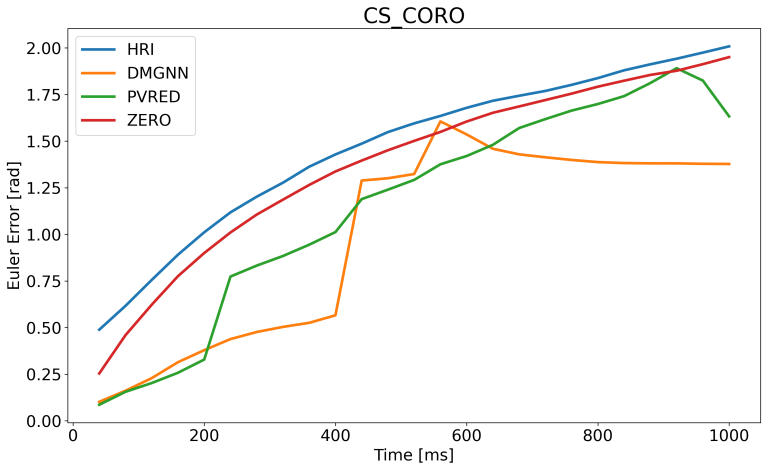
I risultati della metrica della Differenza degli Angoli Articolari in Figura 14b portano a delle considerazioni simili a quella degli Angoli di Eulero. Tale risultato è in linea con quanto atteso, in quanto entrambe analizzano gli errori angolari su tutti i giunti, anche se con rappresentazioni matematiche differenti. Anche in questa analisi la rete che performa meglio risulta essere DMGNN, mentre PVRED e HRI sono più accurate nella stima rispetto allo Zero-Velocity solo dopo alcuni frame di previsione.

Abbastanza differenti sono invece i risultati dell'Errore di Posizione in Figura 14c. Nei primi 200ms di previsione, ovvero nei primi 5 frame, le reti con risultati migliori sono PVRED e DMGNN, tuttavia, in tutti i frame successivi, l'errore minore è quello dell'architettura HRI. Tutte e 3 le reti hanno comunque prestazioni superiori allo Zero-Velocity in quasi tutti i frame, con l'unica eccezione dei primi due frame di HRI.

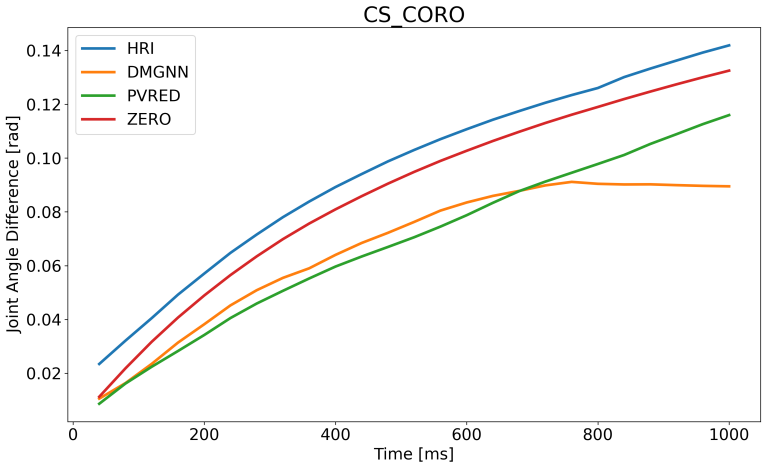
L'analisi delle 3 metriche suggerisce quindi un quadro complesso, le cui diverse sfaccettature portano a conclusioni diverse a seconda del campo d'applicazione in cui si intende utilizzare la previsione. Se si intende concentrarsi principalmente sulla collaborazione, magari con un ampio orizzonte di previsione, allora la rete più adatta sarà HRI poichè è quella che ottiene risultati migliori nell'Errore di Posizione. È infatti questa la metrica che evidenzia maggiormente la precisione nella stima della posizione assoluta delle estremità del corpo come le mani. Viceversa, se il punto focale dell'applicazione è ottenere una posa in cui gli angoli siano il più precisi possibile, allora la scelta più corretta sarebbe DMGNN.

5.3 RISULTATI SU CS_CORO

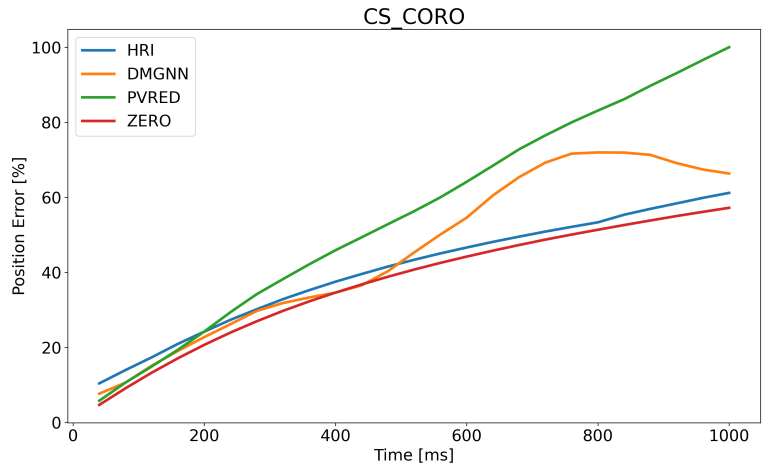
I risultati ottenuti sul nuovo dataset CS_CORO sono illustrati in Figura 15. Osservando in particolare le Figure 15a e 15b, ovvero quelle relative alle metriche degli angoli, si nota che sia PVRED che DMGNN effettuano delle previsioni migliori dello Zero-Velocity. Questo è un ottimo risultato, poichè dimostra che le due architetture riesco-



(a) Errore degli Angoli di Eulero su CS_CORO.



(b) Differenza degli Angoli Articolari su CS_CORO.



(c) Errore di Posizione su CS_CORO.

Figura 15: Risultati di DMGNN, HRI, PVRED e Zero-Velocity su CS_CORO.

no a generalizzare bene i movimenti e consentono di effettuare delle buone previsioni anche su dati che non hanno osservato né durante l'allenamento né nella fase di validazione.

La situazione è invece diversa se si considera l'Errore di Posizione in Figura 15c. Secondo questa metrica, nessuna delle 3 architetture esaminate riesce a superare le prestazioni del modello Zero-Velocity. DMGNN, tutttavia, riesce ad eguagliarne i risultati per previsioni su orizzonti temporali brevi ovvero fino a 400ms.

Tali risultati sono compatibili con quanto atteso e possono essere meglio compresi se si considerano i seguenti fattori:

- Queste reti neurali non sono state addestrate su azioni di collaborazione con dei cobot con diversi oggetti e ostacoli nella zona di lavoro, ma su azioni generiche in ambienti privi di elementi e di impedimenti.
- Data la tipologia delle azioni registrate, è naturale che l'operatore rimanga in posizione quasi statica in alcune fasi. Degli esempi possono essere la transizione tra un azione e la successiva o l'attesa del cobot se in quella determinata circostanza quest'ultimo sta ancora effettuando delle operazioni.

Nelle situazioni appena descritte, il modello Zero-Velocity ha un vantaggio intrinseco ed è quindi normale che ottenga dei buoni risultati nelle varie metriche.

I risultati ottenuti su CS_CORO portano quindi a conclusioni differenti a seconda del campo d'applicazione d'interesse, esattamente come successo per l'analisi effettuata su AMASS. Se l'utilizzo principale di queste architetture si basa sugli angoli dei giunti allora PVRED e DMGNN risultano efficaci senza necessità di ulteriori modifiche anche in contesti differenti da quelli presentati durante l'allenamento. Tuttavia, dato che il punto focale di questa tesi è la robotica collaborativa, i risultati ottenuti sulla metrica di maggiore interesse per questo campo, ovvero l'Errore di Posizione, indicano che esistono ancora margini di miglioramento considerevoli.

5.4 SVILUPPI FUTURI

Alcune possibilità per migliorare i risultati ottenuti sono le seguenti:

- *Fine-tuning:* il *fine-tuning* è una tecnica nell'ambito del *machine learning* che consiste nel regolare o ottimizzare un modello preaddestrato per adattarlo a una specifica attività o dominio. In pratica, si parte da un modello già addestrato su una grande quantità di dati generici come AMASS e lo si adatta ad un contesto più specifico come quello della previsione nella robotica collaborativa. In questo modo diventa possibile migliorare le

prestazioni del modello di base senza la necessità di addestrare un modello da zero, operazione che risulta molto costosa dal punto di vista della generazione dei dati utilizzabili in input.

- *Addestramento su CS_CORO:* se si regista una maggiore quantità di soggetti e di azioni collaborative diventa possibile ottenere i dati necessari non solo per la valutazione delle reti ma anche per il loro addestramento. Utilizzare delle reti addestrate sulla tipologia dei dati d'interesse consente infatti di ottenere delle previsioni molto più accurate sono ogni punto di vista, non solo per quanto riguarda l'Errore di Posizione.

- *Aggiunta dell'informazione semantica:* questa strategia consiste nell'aggiungere in input alle reti anche l'informazione *semantica*, ovvero la descrizione della posizione degli oggetti e degli ostacoli che sono presenti nell'ambiente. In questo modo diventa possibile per le reti neurali mettere in relazione i movimenti della persona assieme agli elementi presenti. L'introduzione di questi ulteriori dati consente quindi di prevedere i movimenti futuri dell'operatore basandosi sia sulle sue pose passate che sugli oggetti con cui sta interagendo o con cui è più probabile che interagirà. Come esempio, si può considerare che se l'operatore ha una vite in una mano è molto probabile che l'azione successiva sarà raccogliere un cacciavite o un trapano. Questa strategia risulta particolarmente efficace soprattutto in ambienti strutturati come nel caso di una stazione di assembleggio, consentendo di ridurre l'incertezza sulle azioni future dell'operatore.

Tali strategie non sono mutualmente escludenti, perciò mediante una combinazione di queste soluzioni è possibile migliorare considerevolmente i risultati di stima delle pose future. Il risultato migliore si otterrà con una rete addestrata completamente su azioni di collaborazione con cobot che tenga anche conto dell'informazione semantica.

CONCLUSIONI

Nel corso di questa tesi sono state analizzate diverse architetture di reti neurali per la predizione del movimeno umano, concentrandosi sull'ambito della robotica collaborativa. L'obiettivo è stato comprendere se queste architetture sviluppate per predire dei movimenti generici fossero in grado di effettuare delle previsioni efficaci anche in contesti in cui avviene una collaborazione tra uomo e robot.

In particolare, sono state individuate le reti DMGNN, HRI e PVRED e ne è stato analizzato nel dettaglio il funzionamento. La scelta di DMGNN è dovuta al suo particolare approccio basato sulla modellizzazione del corpo umano tramite grafi. La rete HRI è stata selezionata per via dell'utilizzo del meccanismo dell'attenzione, soluzione alla base dell'architettuta Transformer che sta riscuotendo notevole successo in svariate applicazioni. Infine, PVRED è stata scelta perchè basata su una RNN, ovvero l'approccio maggiormente utilizzato in letteratura per predire il movimento umano. Tuttavia, PVRED si differenzia rispetto alle altre RNN per la predizione del movimento poiché sfrutta come input sia le pose che la velocità delle pose stesse.

Tali architetture sono state inoltre adattate per consentire dei paragoni efficaci su 2 dei più importanti dataset di movimenti umani: H3.6M e AMASS. H3.6M è stato scelto per via del suo ampio utilizzo nelle reti che si occupano di predizione del movimento. AMASS è stato utilizzato per la sua enorme quantità di dati che permette l'addestramento di modelli di previsione più avanzati.

Inoltre, sono state definite le metriche più adatte per valutare i risultati sotto diversi punti di vista: l'Errore degli Angoli di Eulero, la Differenza degli Angoli Articolari e l'Errore di Posizione. Le prime due forniscono importanti informazioni sulla qualità della previsione di ogni giunto del corpo, mentre con la terza, è possibile valutare l'accuratezza del posizionamento spaziale del corpo, un fattore di notevole importanza nella robotica collaborativa. È stato anche introdotto come baseline il modello Zero-Velocity, così da consentire il confronto tra i risultati assoluti dei 3 sistemi con un punto di riferimento standard utilizzato nella letteratura.

Oltre ad effettuare le analisi delle reti su H3.6M e AMASS, in questa tesi è stato raccolto un nuovo dataset completamente focalizzato sulla collaborazione uomo-robot in un contesto industriale: CS_CORO. Grazie a questo dataset è stato possibile verificare in quale misura le 3 reti neurali considerate fossero efficaci per prevedere questo tipo di dati. Poichè le informazioni contenute in questo dataset sono state registrate tramite una grande varietà di sensori e strumenti all'avan-

guardia, le possibilità di utilizzo di CS_CORO si possono estendere ben oltre gli scopi di questa tesi.

I risultati ottenuti su H3.6M e su AMASS portano a concludere che lo specifico campo d'interesse in cui si intende utilizzare le reti neurali per la previsione del movimento umano è una variabile di fondamentale importanza: reti come DMGNN e PVRED ottengono risultati migliori nelle metriche relative agli angoli del corpo, mentre HRI ottiene ottimi risultati nella previsione delle posizioni assolute.

A loro volta, i risultati ottenuti su CS_CORO si interpretano in modo diverso a seconda del campo d'applicazione e quindi delle metriche più rilevanti. Analizzando l'errore degli Angoli di Eulero e la Differenza degli Angoli Articolari si osserva che DMGNN e PVRED sono le due reti più efficaci e riescono a battere il modello Zero-Velocity anche su dati per cui non sono state allenate. Sono quindi due reti che riescono a generalizzare bene i movimenti appresi durante l'addestramento.

La situazione è differente quando si considera l'Errore di Posizione: da questo punto di vista, nessuna delle reti analizzate riesce a ottenere dei risultati migliori del modello Zero-Velocity, rimanendovi tuttavia molto vicine. Ciò è particolarmente incoraggiante considerando che nessuna delle reti è stata addestrata su questo tipo di dati, evidenziando che, come ipotizzato, i modelli allenati su dati generici non riescono a fornire risultati ottimali per dati specifici sulla robotica collaborativa. Tali risultati, però, rimangono comparabili con l'output ottenuto con il modello Zero-Velocity indicando che i modelli testati hanno il potenziale per gestire la complessità intrinseca della collaborazione tra uomo e robot in un contesto industriale.

In prospettiva futura si possono percorrere diverse strade per proseguire lungo la direzione tracciata da questa tesi e potenzialmente migliorare i risultati ottenuti. Oltre a verificare il funzionamento di altre promettenti architetture, come ad esempio *A Spatio-temporal Transformer for 3D Human Motion Prediction* [2] o *Bidirectional Transformer GAN for Long-term Human Motion Prediction* [31], è inoltre possibile testare alcune tecniche di machine learning che hanno fornito ottimi risultati in altri settori per gestire dataset con disponibilità di dati limitate.

Le due principali alternative sono legate a *data augmentation* e *transfer learning*. La prima tecnica mira ad incrementare i dati in maniera sintetica fino ad ottenere sufficienti esempi da rendere robusto l'allenamento da zero di un modello completo. La seconda tecnica ha l'obiettivo di ottimizzare i modelli pre-addestrati su dati generici e renderli efficaci nello specifico contesto del dataset più limitato, nel nostro caso della robotica collaborativa.

Un ulteriore aspetto da indagare è l'ampliamento a più soggetti e più azioni dello studio condotto in questa tesi. In tal modo diventa

possibile comprendere al meglio quali siano gli aspetti che caratterizzano l'anticipazione del movimento nell'ambito della robotica collaborativa e modificare di conseguenza i modelli presi in esame o crearne di nuovi.

Infine, un'ultima strategia che si può adottare consiste nell'aggiungere in ingresso alle reti le informazioni semantiche relative agli oggetti che vengono manipolati e agli ostacoli che sono presenti nell'ambiente. In questo modo diventa possibile per le reti neurali mettere in relazione i movimenti della persona assieme agli elementi presenti. I movimenti futuri vengono così predetti basandosi sia sulle pose passate che sugli oggetti con cui la persona sta interagendo o con cui interagirà.

In conclusione, grazie alle analisi effettuate in questa tesi, al dataset collaborativo raccolto e considerando le ulteriori prospettive di ricerca presentate, il mio lavoro ha aperto la strada per migliorare le prestazioni dei modelli di previsione del movimento esistenti consentendo così una migliore collaborazione uomo-robot, un aumento della sicurezza e una maggior efficienza negli ambienti industriali.

BIBLIOGRAFIA

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7143–7152, 2019.

[2] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. *2021 International Conference on 3D Vision (3DV)*, pages 565–574, 2020.

[3] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, Ding Liu, Jing Liu, and Nadia Magnenat Thalmann. Learning progressive joint propagation for human motion prediction. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 226–242, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58571-6.

[4] Carnegie Mellon University. CMU MoCap Dataset. URL http://mocap.cs.cmu.edu.

[5] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[6] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F. Troje. Movi: A large multipurpose motion and video dataset, 2020.

[7] Rexroth Bosch Group. Ergonomics guidebook for manual production systems. URL https://dc-gb.resource.bosch.com/media/gb/products_12/product_groups_2/assembly_technology_3/manual_production_systems/ergonomics_1/3842525794_2015_05_EN_Ergomomieratgeber_Media.pdf.

[8] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1325–1339, jul 2014.

[9] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. *CoRR*, abs/1805.00655, 2018.

[10] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton-based human motion prediction. *CoRR*, abs/2003.08802, 2020.

[11] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

[12] Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 33(6):220:1–220:13, November 2014. doi: 10.1145/2661229.2661273. URL http://doi.acm.org/10.1145/2661229.2661273.

[13] Eyes JAPAN Co. Ltd. Eyes japan dataset. URL http://mocapdata.com/.

[14] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5441–5450, October 2019. doi: 10.1109/ICCV.2019.00554.

[15] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. Unifying representations and large-scale whole-body motion databases for studying human motion. *IEEE Transactions on Robotics*, 32(4):796–809, 2016.

[16] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, 2020.

[17] Julieta Martinez, Michael Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *ArXiv*, 07 2017. doi: 10.1109/CVPR.2017.497.

[18] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. *CoRR*, abs/1705.02445, 2017.

[19] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[20] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[21] Payam Nikdel, Mohammad Mahdavian, and Mo Chen. Dmmgan: Diverse multi motion prediction of 3d human joints using attention-based generative adverserial network, 2022.

[22] Archive of Motion Capture As Surface Shapes. Transitions. URL https://amass.is.tue.mpg.de/.

[23] Simon Fraser University & National University of Singapore. Sfu motion capture database. URL https://mocap.cs.sfu.ca/.

[24] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731, June 2021.

[25] Matt Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *2017 British Machine Vision Conference (BMVC)*, 2017.

[26] Ohio State University. Advanced computing center for the arts and design. URL https://accad.osu.edu/research/motion-lab/mocap-system-and-data.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[28] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

[29] Hongsong Wang and Jiashi Feng. VRED: A position-velocity recurrent encoder-decoder for human motion prediction. *CoRR*, abs/1906.06514, 2019.

[30] Jianjing Zhang, Hongyi Liu, Qing Chang, Lihui Wang, and Robert X. Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP Annals*, 69(1):9–12, 2020. ISSN 0007-8506. doi: https://doi.org/10.1016/j.cirp.2020.04.077. URL https://www.sciencedirect.com/science/article/pii/S0007850620300998.

[31] Mengyi Zhao, Hao Tang, Pan Xie, Shuling Dai, Nicu Sebe, and Wei Wang. Bidirectional transformer gan for long-term human motion prediction. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(5), apr 2023. ISSN 1551-6857. doi: 10.1145/3579359. URL https://doi.org/10.1145/3579359.