



# Università di Catania

Dipartimento di Matematica ed Informatica  
Corso di laurea Magistrale in Informatica  
A.A. 2022-2023

---

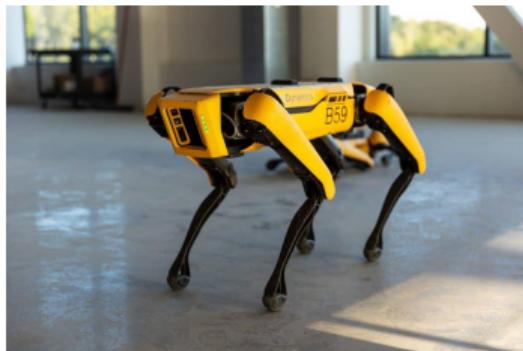
**Danilo Leocata**

*Sviluppo di Sistemi di Navigazione Robotica in Ambienti Dinamici basati su Deep Reinforcement Learning*

Relatore: Prof. **Giovanni Maria Farinella**

Correlatore: Dott. **Marco Rosano**

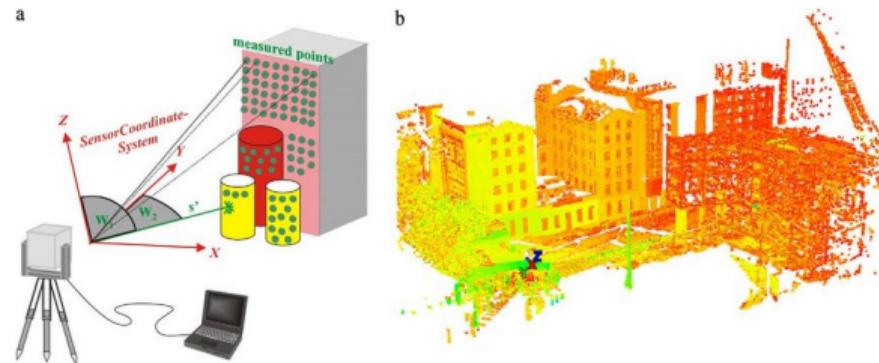
# Piattaforme robotiche



# Caso d'uso

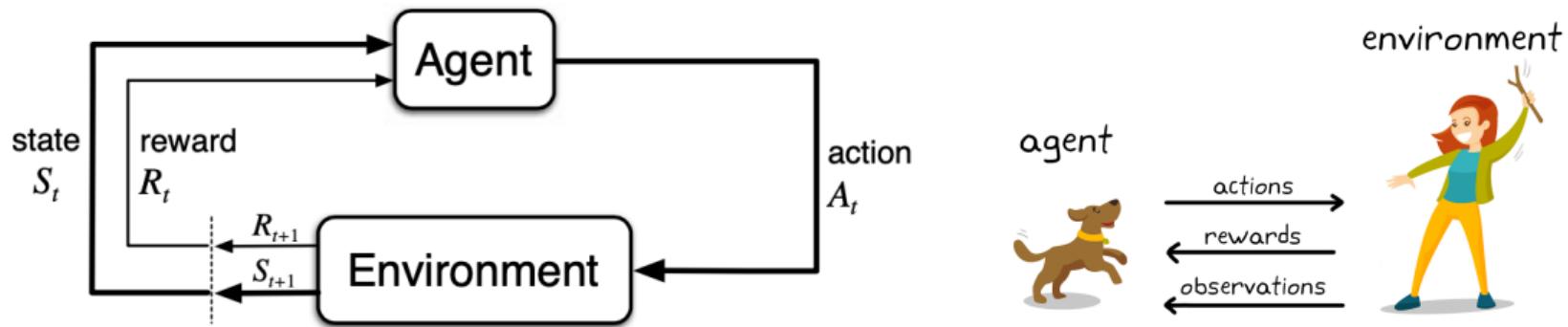


**Figure 1:** Robot per il quale si vuole sviluppare il modello di navigazione, evidenziato in rosso il sensore LiDAR equipaggiato



**Figure 2:** Funzionamento sensore LiDAR: (a) il processo di scansione laser per la misurazione di punti 3D. (b) Un esempio di dati scansionati al laser di un edificio in costruzione

# Reinforcement Learning



- $R_t$ : ricompensa, ottenuta dall'agente, *positiva* o *negativa*
- $\sum_{t=0}^T R_t$ : sommatoria delle ricompense ottenute dall'agente durante l'episodio di navigazione

<https://github.com/vita-epfl/CrowdNav>

## Pro

- OpenSource;
- Modelli per effettuare il training già implementati;
- Cinematica di movimento del robot di destinazione già implementata;
- Codice non complesso e requisiti di sistema minimi;

## Contro

- Sensore LiDAR non supportato;
- Dimensione fissa dei limiti dell'ambiente;
- Import planimetrie ed ostacoli non previsto;

[Figure 3: Simulazione di un episodio di test di CrowdNav allo stato originale.](#)

# Modifiche apportate

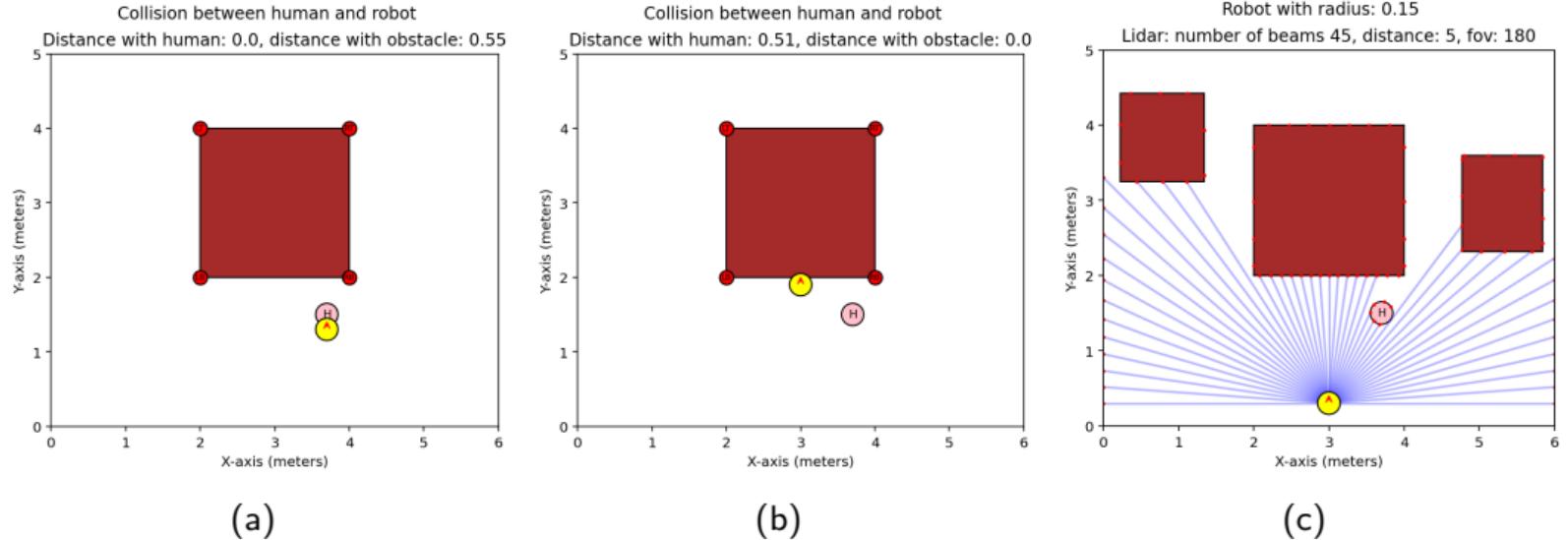
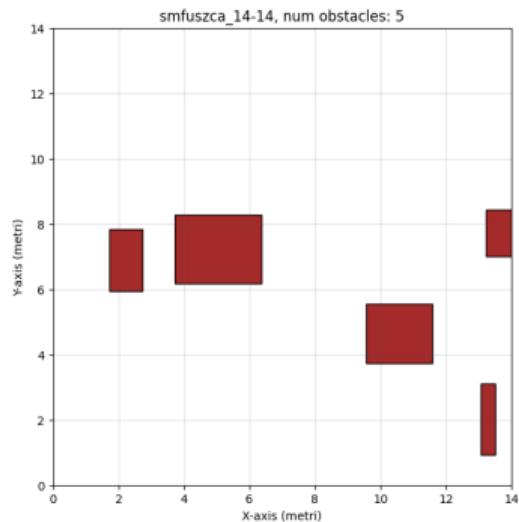
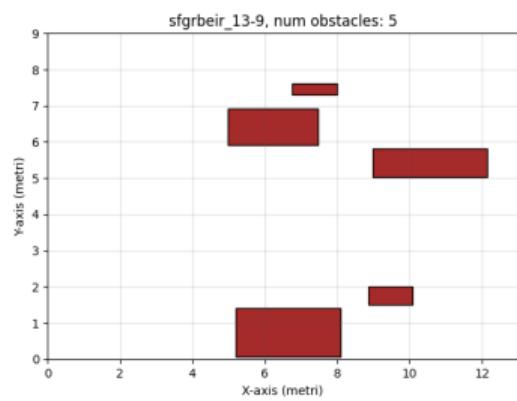
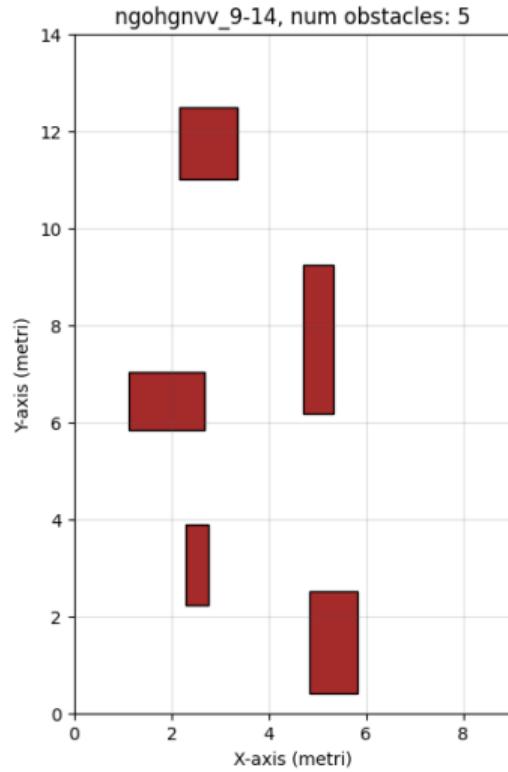


Figure 4: (a) e (b) import di un ostacolo ed esempio di collisione (c) rendering di robot con LiDAR

# Configurazione ambiente: dinamico



# Configurazione ambiente: planimetrie reali

Airport.glb\_0.1.png

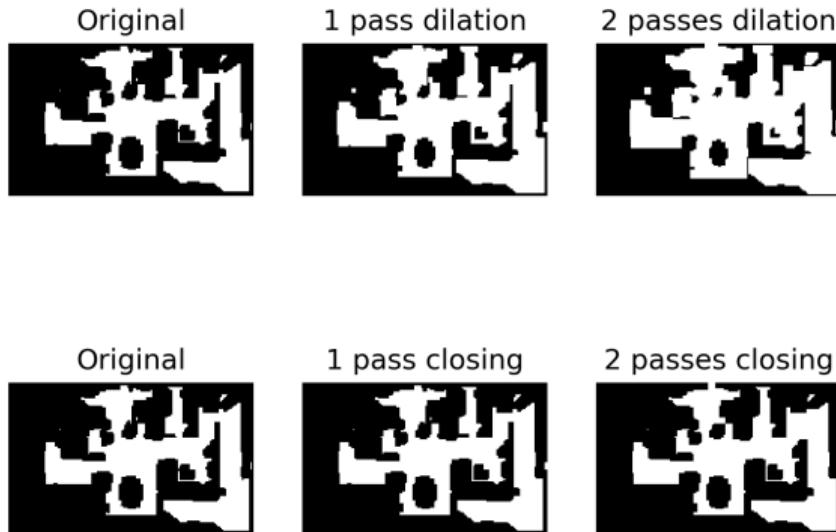


Figure 6: Immagine di partenza di un ambiente preso dal dataset Gibson

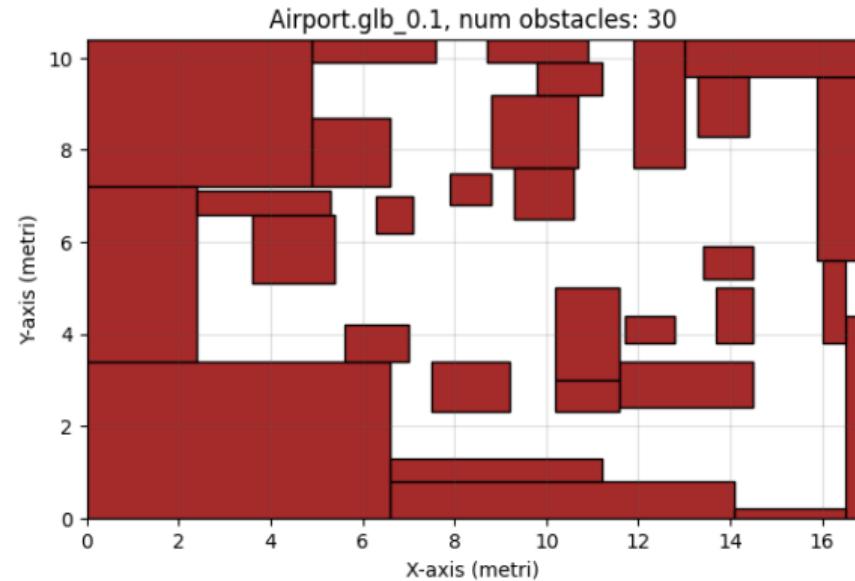


Figure 7: Ambiente dell'immagine a sinistra ricostruito per farlo funzionare nel simulatore

---

## Algorithm 1: CADRL (Coll. Avoidance with Deep RL)

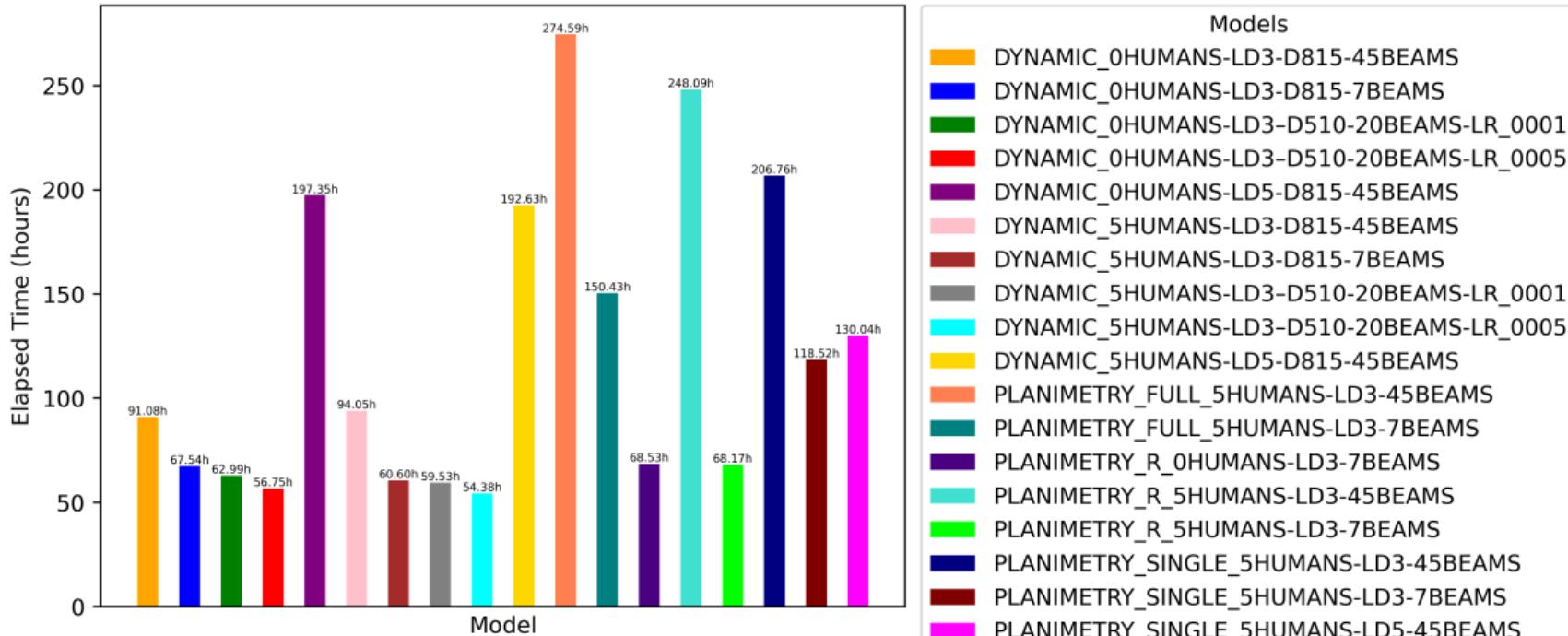
---

- 1 **Input:** value network  $\mathbf{V}(\cdot; \mathbf{w})$
- 2 **Output:** trajectory  $\mathbf{s}_{0:t_f}$
- 3 **while** not reached goal **do**
- 4     update  $t$ , receive new measurements  $\mathbf{s}_t, \tilde{\mathbf{s}}_t^o$
- 5      $\hat{\mathbf{v}}_t \leftarrow \text{filter}(\tilde{\mathbf{v}}_{0:t})$
- 6      $\hat{\mathbf{s}}_{t+1}^o \leftarrow \text{propagate}(\tilde{\mathbf{s}}_t^o, \Delta t \cdot \hat{\mathbf{v}}_t)$
- 7      $A \leftarrow \text{sampleActions}()$
- 8      $\mathbf{a}_t \leftarrow \text{argmax}_{\mathbf{a}_t \in A} R(\mathbf{s}_t^{jn}, \mathbf{a}_t) + \bar{\gamma} V(\hat{\mathbf{s}}_{t+1}, \hat{\mathbf{s}}_{t+1}^o)$   
      where  $\bar{\gamma} \leftarrow \gamma^{\Delta t \cdot v_{pref}}$ ,  $\hat{\mathbf{s}}_{t+1} \leftarrow \text{propagate}(\mathbf{s}_t, \Delta t \cdot \mathbf{a}_t)$
- 9 **return**  $\mathbf{s}_{0:t_f}$

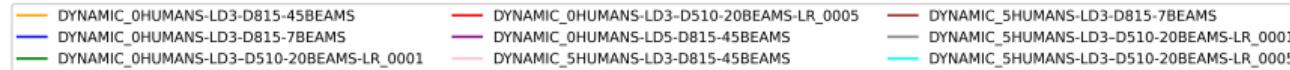
---

# Esperimenti

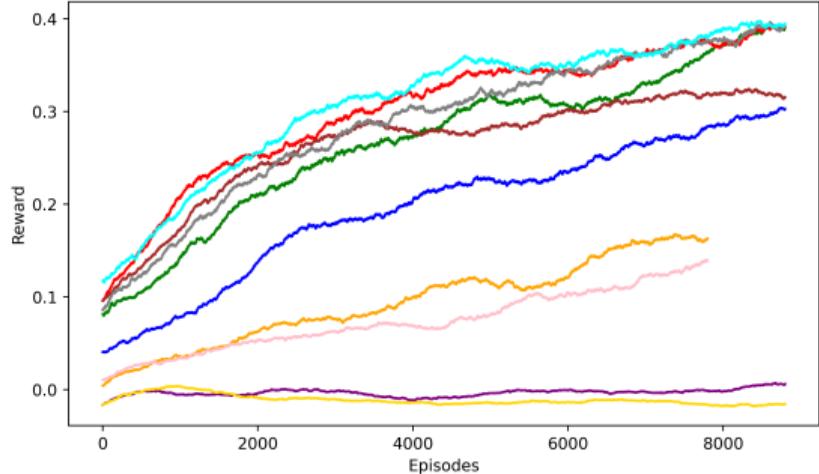
## Elapsed Time for Different Models



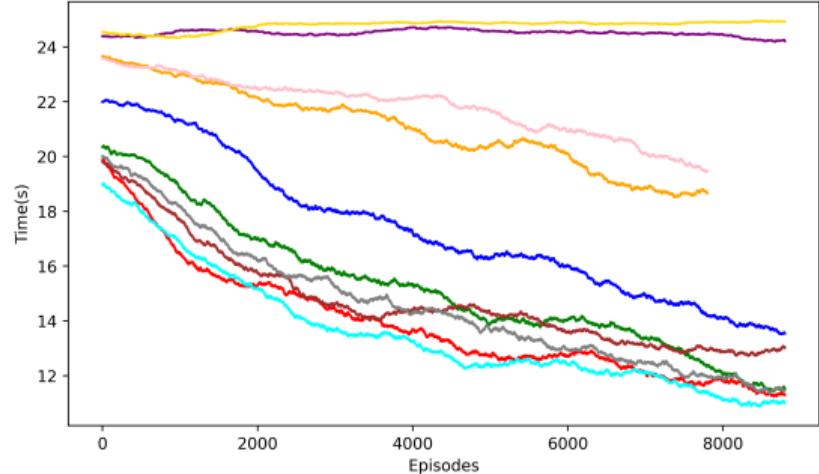
# Fase di training



Cumulative Discounted Reward (train)



Robot's Time to Reach Goal (train)



# Esempi di navigazione in ambienti dinamici

# Esempi di navigazione in ambienti con planimetrie

# Conclusioni e sviluppi futuri

- Introduzione di una penalità nel calcolo della reward in caso di stallo;
- Ridurre il tempo necessario per il calcolo del vettore del LiDAR;
- Rendere proporzionale la dimensione del modello alla dimensione dell'input ed utilizzare modelli più complessi come LSTM-RL e SA-CADRL;
- Incrementare il numero di episodi durante la fase di addestramento;
- Implementazione esecuzione in parallelo del training su GPU;

# Grazie per l'attenzione!