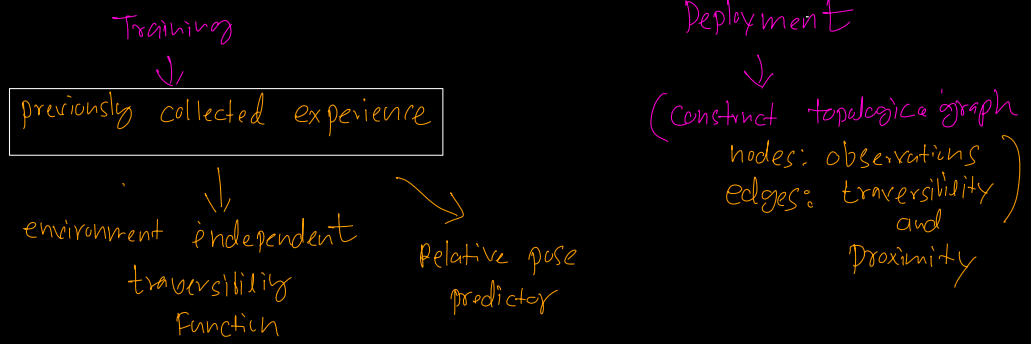


At each time step (t): localize current (Ot) and goal (Og) observations in the graph and Follow the best path to (G) which is generated by a graph search algorithm (output waypoints at each

Definitions

Traversability Function $T(o_i, o_j) \in \mathbb{R}^+$

① whether any controller can navigate from o_i to o_j successfully. (number of time steps required by a controller to navigate from o_i to o_j)

② two ways to learn supervised and temporal difference learning

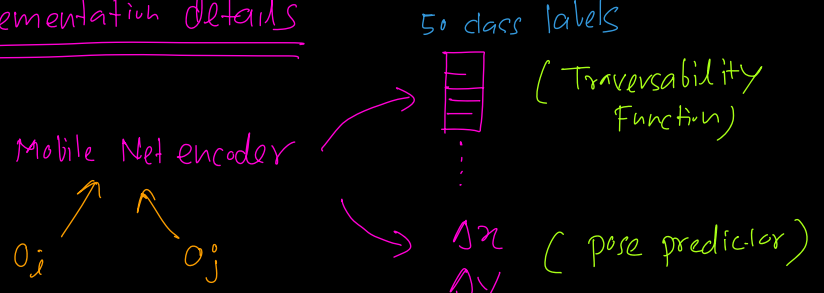
Supervised For a trajectory Regress From $(o_i, o_j) \rightarrow j - i$ For the controller that performed this Run (overestimate shortest path)

Temporal difference -> gives the shortest path.

Pose predicter

$(o_i, o_j) \rightarrow \Delta P_{ij}$ (From image observation to relative pose using onboard odometry.

Implementation details



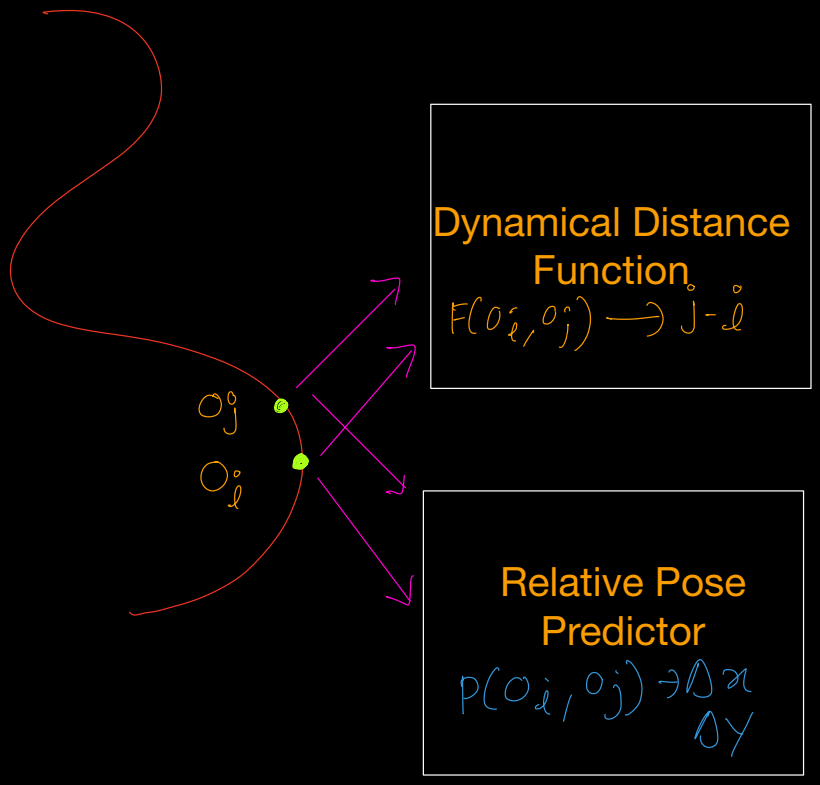
Topological graph

nodes: All the observations in the Trajectories

edges: For every pair of nodes add an edge with Traversability and pose prediction according to the learned Functions above.

Planning in the graph

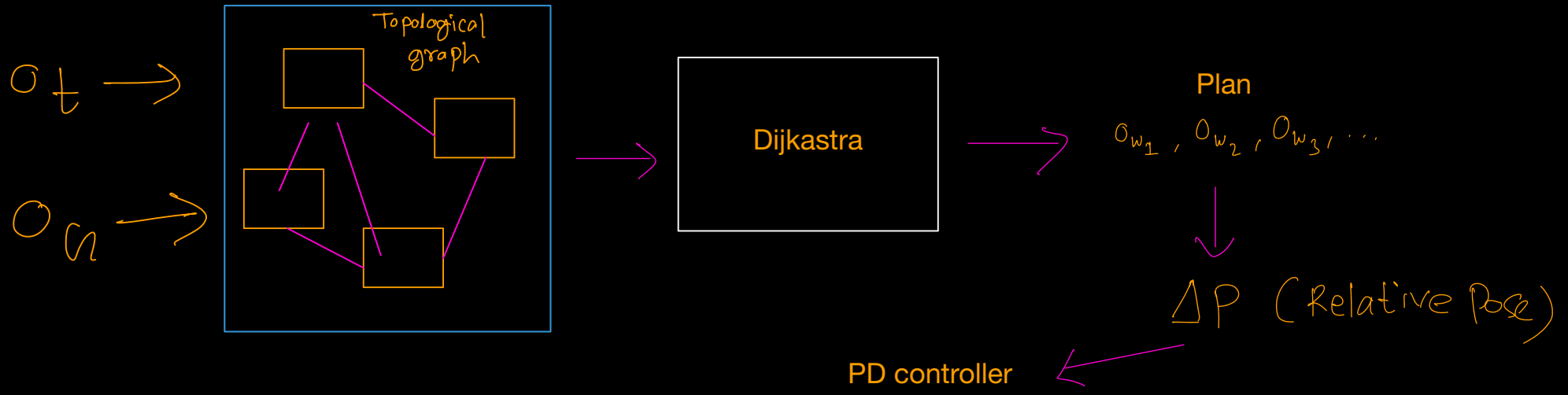
Add edges corresponding to current and goal observations and use dijkstra.



Topological Graph

Nodes: All the observations in the Trajectories

Edges: For every pair of nodes add an edge with Traversability and pose prediction according to the learned Functions above.



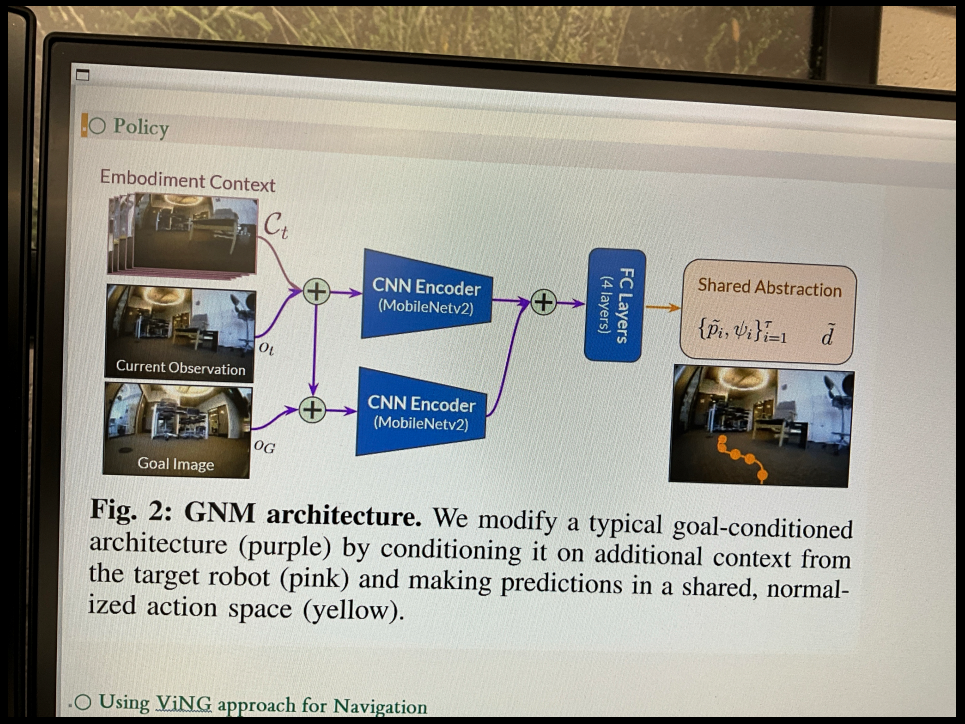


Fig. 2: GNM architecture. We modify a typical goal-conditioned architecture (purple) by conditioning it on additional context from the target robot (pink) and making predictions in a shared, normalized action space (yellow).

Topological Graph

Nodes: All the observations in the Trajectories

Edges: For every pair of nodes add an edge with Traversability and pose prediction according to the learned Functions above.