IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Cooperative Grasping for Collective Object Transport in Constrained Environments

David Alvear, George Turkiyyah, and Shinkyu Park

***Abstract*—We propose a novel framework for decision-making in cooperative grasping for two-robot object transport in constrained environments. The core of the framework is a *Conditional Embedding (CE)* model consisting of two neural networks that map grasp configuration information into an embedding space. The resulting embedding vectors are then used to identify feasible grasp configurations that allow two robots to collaboratively transport an object. To ensure generalizability across diverse environments and object geometries, the neural networks are trained on a dataset comprising a range of environment maps and object shapes. We employ a supervised learning approach with negative sampling to ensure that the learned embeddings effectively distinguish between feasible and infeasible grasp configurations. Evaluation results across a wide range of environments and objects in simulations demonstrate the model's ability to reliably identify feasible grasp configurations. We further validate the framework through experiments on a physical robotic platform, confirming its practical applicability.**

***Index Terms*—Multi-Robot Systems, Mobile Manipulation, Grasp Planning, Deep Learning, Motion Planning.**

I. Introduction

COLLECTIVE object transport is a challenging problem that involves determining the optimal grasp strategy—specifically, where and how each robot should grasp the object—to enable collaborative manipulation and transportation to a target location. This task demands precise coordination to ensure the object is securely grasped and that a feasible trajectory to the target destination exists. The complexity of identifying suitable grasp configurations increases significantly with the geometric intricacy of both the object and the surrounding environment. In this work, we propose a learning-based framework for identifying feasible grasp configurations for two robots, aiming to enable reliable object transport in constrained environments (see Fig. 1).

Our proposed framework begins by generating a set of candidate grasp configurations, where each configuration defines a grasp point on the object for each robot and corresponding robot position required to execute the grasp. To identify viable configurations for object transport, we propose a *Conditional Embedding (CE)* model. This model comprises two neural networks that project grasp configuration information into a shared embedding space, enabling the evaluation of candidate pairs through a similarity metric. The model is trained to assign high similarity scores to feasible pairs while penalizing infeasible pairs—even in the presence of class imbalance in the training data.

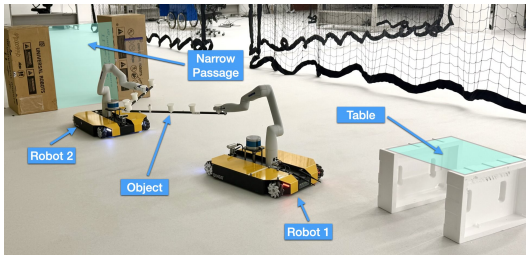


Fig. 1: Object transport using two mobile manipulators. Each robot selects a grasp on the object to collaboratively move it from its initial location on the table to a designated destination, navigating through a narrow passage.

We validate the effectiveness of the proposed framework through extensive physics-based simulations and demonstrate its practical applicability on a physical robotic platform. While our primary contribution focuses on the design and validation of two-robot object transport, we also illustrate how the same framework can be extended to scenarios involving more than two robots by generalizing the CE model's training objective.

Related Work: A wide range of studies on multi-robot object manipulation and transportation are relevant to our work. For example, Eoh *et al.* [1] investigated push-pull strategies for transporting heavy objects using coordinated multi-robot formations. In grasp planning, Vahrenkamp *et al.* [2] introduced a multi-robot RRT-based planner that runs parallel bimanual inverse kinematics RRT (IK-RRT) instances, guided by inverse reachability maps to probabilistically sample feasible grasp pairs. Grasp candidates are filtered using a wrench-based threshold before computing collision-free trajectories. Similarly, Tariq *et al.* [3] proposed a grasp planner that minimizes a wrench-based metric to facilitate effective load sharing in collaborative manipulation. Muthusamy *et al.* [4]–[6] developed a decentralized cooperative grasping framework that generates grasp candidates and selects among them using a wrench-based quality metric. Liu et al. [7] introduced a planning framework to enable multiple mobile manipulators to perform complex flipping manipulation. Additionally, Nachum *et al.* [8] proposed a hierarchical policy framework that separates control into two levels: a low-level policy for individual robot locomotion and a high-level policy for coordinating multi-robot manipulation.

Existing studies on multi-robot object transport can be broadly categorized into decentralized/distributed control,

Manuscript received: August 21, 2025; Revised October 20, 2025; Accepted December 22, 2025.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by funding from King Abdullah University of Science and Technology (KAUST), and the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI).

The authors are with the Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia. {david.alvear, george.turkiyyah, shinkyu.park}@kaust.edu.sa

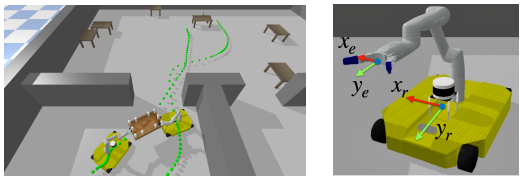
Digital Object Identifier (DOI): see top of this page.

optimization-based planning, and learning-based methods. In decentralized approaches, Chen *et al.* [9] proposed a swarm-based approach in which robots uniformly surround an object and use local sensing to transport it toward a target, guided by the "occlusion" of the target location. Habibi *et al.* [10] developed a distributed control strategy using local communication and centroid estimation, enabling dexterous transport via four motion controllers. Similarly, Farivarnejad *et al.* [11] implemented a sliding-mode controller for cooperative payload transport that relies solely on local velocity and heading measurements. Savino et al. [12] introduced a decentralized, consensus-based strategy for multi-robot formation control, using dual quaternions for a unified representation of position and orientation to enable decentralized control of end-effectors. While effective in decentralized coordination, these methods do not address optimal robot placement for grasping and transporting objects in constrained environments.

In optimization-based planning, Alonso-Mora *et al.* [13] introduced a formation control framework that enables object transport in environments with both static and dynamic obstacles. Koung *et al.* [14] employed hierarchical quadratic programming (HQP) for coordinated object transport under rigid formation constraints. Vlantis *et al.* [15] proposed a hierarchical space decomposition method to facilitate object transport in cluttered planar environments. Kennel et al. [16] presented a bi-level optimization system for cooperative transport with non-holonomic mobile manipulators. Their lower-level controller calculates contact forces to ensure stability, which are used by the higher-level planner to formulate constraints for tip-over avoidance. While these approaches effectively account for environmental constraints, they typically rely on a fixed or limited set of initial robot formations and assume fixed object geometries. Consequently, they do not address the challenge of computing feasible grasp configurations for objects with diverse shapes and in environments with complex spatial constraints.

In learning-based approaches, Eoh and Park [17] presented a deep reinforcement learning framework for multi-robot object transport, using a region-growing curriculum and a single-to-multiple-robot training scheme, where robots interact with the object via pushing in a discrete action space. Zhang *et al.* [18] proposed a decentralized control strategy where each robot employs a Deep Q-Network to translate sensory inputs into control commands for transporting a long rod through a narrow doorway. While effective for learning control policies, these methods typically assume fixed robot formations and do not adapt to variations in object geometry and environmental layout, limiting their applicability to more complex scenarios involving adaptive grasping.

Distinct from prior work, our study focuses on identifying feasible grasp configurations for mobile manipulators operating in constrained environments—specifically during object transport through narrow passages. To address this challenge, we propose the CE model trained via supervised learning on labeled data generated from a physics-based simulator and a negative sampling technique, which significantly accelerates the training. The model directly predicts feasible grasp configurations with high accuracy, avoiding exhaustive candidate evaluation and substantially reducing the computational overhead of trajectory planning.



(a) (b)

Fig. 2: (a) A collective object transport task involving two mobile manipulators navigating a narrow passage while carrying a large object that requires both robots to transport. (b) A mobile manipulator used in this work.

Paper Organization: In Section II, we introduce the preliminaries and formalize the problem of the cooperative grasping for object transport. Section III presents the proposed framework and introduces the CE model as a solution to the problem. In Section IV, we provide simulation results to evaluate our framework and demonstrate its applicability on a physical robot platform. Section V concludes the paper.

II. Preliminaries and Problem Description

The environment includes walls and randomly placed tables, which act both as obstacles and as platforms on which the object is initially positioned. It also features narrow passages (Fig. 2(a)), introducing additional constraints on navigation and object transport. The robot (Fig. 2(b)) is equipped with an omnidirectional mobile base operating in a 2D plane, with position denoted by (x_r, y_r). It includes a manipulator with a gripper, whose end-effector position is given by (x_e, y_e, z_e). During transport, the gripper operates at a fixed height $z_e = 0.45\ m$, allowing control only in the x- and y-directions. The object is allowed to rotate only about its yaw axis.

A. Definitions

Environment Map and Object Geometry: Let $\mathbb{M} \subset \mathbb{R}^2$ denote the map of the environment, partitioned into the free space $\mathbb{M}_{\text{free}}$ and the obstacle region $\mathbb{M}_{\text{obs}}$. The object's geometry is modeled as a 2D polygon from a top-down view, defined by a set of vertices $\mathbb{V} = \{v^1, \cdots, v^h\}$ that describe its boundary. Each vertex $v \in \mathbb{V}$ lies on the object's perimeter.

Grasp Configurations: Let $\mathbb{F} = \{F^1, \cdots, F^n\} \subset \mathbb{R}^2$ denote a set of predefined grasp points distributed along the object's boundary, where each $F = (x_g, y_g) \in \mathbb{F}$ defines a feasible grasp point. The number of points n is chosen to ensure sufficient coverage of the object's perimeter, enabling possible grasps from all directions. Given an object with $\mathbb{F}$ as its potential grasp points, we define the set of grasp configurations as $\mathbb{G} = \{G^1, \cdots, G^m\} \subset \mathbb{M}_{\text{free}} \times \mathbb{F}$, where each configuration $G \in \mathbb{G}$ is a tuple $G = (x_r, y_r, x_g, y_g)$. Here, $(x_r, y_r) \in \mathbb{M}_{\text{free}}$ represents the collision-free position of the mobile base from which the end-effector can reach a grasp point $(x_g, y_g) \in \mathbb{F}$. Details on the construction of grasp configuration set are provided in Appendix A.

Feasible Grasp Configurations and Trajectory Planner: Our framework draws inspiration from the Skip-Gram model in natural language processing, which captures the relationship

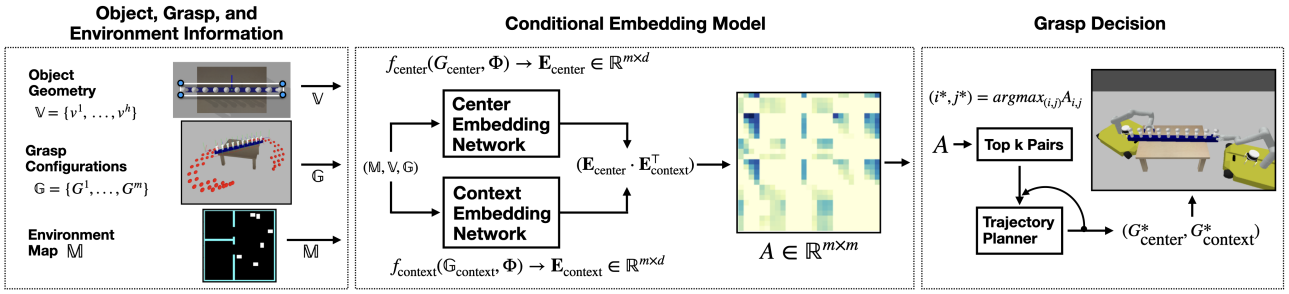


Fig. 3: Grasp configuration pair selection using the CE model: Candidate grasp configurations for two mobile manipulators are generated (left block). Two embedding networks compute an affinity matrix A that scores all possible configuration pairs (middle block). The top-k pairs, ranked by affinity score, are selected and evaluated for feasibility using a trajectory planner (right block).

between a *center* word and its surrounding *context* words. Analogously, we refer to the first robot's grasp as the *center* grasp G_{center} and the second robot's grasp as the *context* grasp G_{context}. We define a binary metric $S(G_{\text{center}}, G_{\text{context}}) \in \{0, 1\}$ to indicate whether two robots can successfully transport an object to its destination using G_{center} and G_{context}. The metric is determined by evaluating the outcome of a trajectory planner, which computes the trajectories of two robots transporting the object, as detailed in Appendix B. Note that $S = 1$ indicates that a feasible transportation trajectory exists, while $S = 0$ denotes failure.

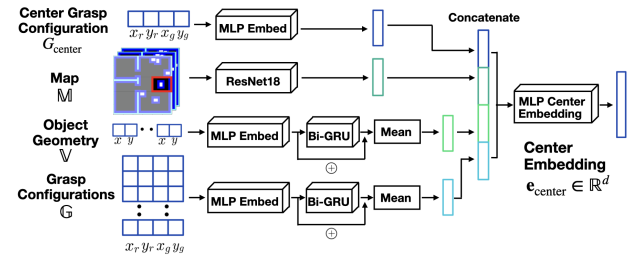
B. Problem Description

Given the tuple $\Phi = (\mathbb{M}, \mathbb{V}, \mathbb{G})$—which encapsulates the environment map ($\mathbb{M}$), object geometry ($\mathbb{V}$), and candidate grasp configurations ($\mathbb{G}$)—the objective is to determine a pair of grasp configurations $(G_{\text{center}}, G_{\text{context}})$ that enable two mobile manipulators to execute an object transport task. We propose a learning-based framework in which a policy π_θ, conditioned on Φ, is trained to predict an affinity matrix $A \in \mathbb{R}^{m \times m}$ with $m = |\mathbb{G}|$, such that $A = \pi_\theta(\Phi)$. The affinity matrix A encodes the feasibility of all possible pairs of grasp configurations for successful object transport. Once the affinity matrix A is obtained, the optimal grasp configuration pair $(G^*_{\text{center}}, G^*_{\text{context}})$ is selected by identifying the index pair (i^*, j^*) that maximizes the affinity score: $(i^*, j^*) = \arg\max_{i,j} A_{ij}$, where A_{ij} denotes the i, j-th entry of A, and $G^*_{\text{center}} = G^{i^*}$ and $G^*_{\text{context}} = G^{j^*}$. Alternatively, the matrix A can be used to retrieve the top-k grasp configuration candidates for downstream evaluation or selection.

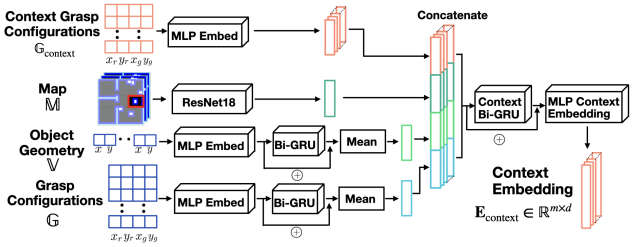
Note that a brute-force approach, which exhaustively evaluates all possible grasp configuration pairs, requires $\frac{m!}{(m-N)!}$ evaluations to identify the optimal pair, where N is the number of robots (with $N = 2$ in our case) and m is the number of candidate grasp configurations. As a result, the computational complexity grows quadratically with m, i.e., $O(m^2)$. This highlights the significant computational burden associated with identifying feasible configuration pairs when the number of candidates is large.

III. Cooperative Grasping Framework

In the proposed framework, the trained CE model takes the input state Φ and generates k candidate grasp configuration pairs. Each pair is then evaluated by the trajectory planner to determine whether a collision-free path for object transport exists. The overall pipeline for identifying feasible grasp configurations using the CE model is illustrated in Fig. 3.



(a)



(b)

Fig. 4: Architectures for (a) the center embedding network and (b) the context embedding network.

A. Conditional Embedding (CE) Model

The CE model consists of two neural networks: the *center embedding* network and *context embedding* network (see Figs. 3 and 4). Each network maps the grasp configuration information into an embedding space $\mathbb{R}^d$, producing a pair of embeddings. Here, d is a configurable parameter that defines the dimensionality of the embedding space. The model is trained such that the embeddings corresponding to a grasp pair $(G_{\text{center}}, G_{\text{context}})$ are positively correlated if the pair is feasible for object transport, i.e., $S(G_{\text{center}}, G_{\text{context}}) = 1$.

Center Embedding Network f_{center}: The center embedding vector $\mathbf{e}_{\text{center}} \in \mathbb{R}^d$ (with unit length) for $G_{\text{center}} \in \mathbb{G}$ is generated by a function f_{center}: $\mathbf{e}_{\text{center}} = f_{\text{center}}(G_{\text{center}}, \Phi)$. The function f_{center} is implemented as a neural network, whose architecture is illustrated in Fig. 4(a). Elements of the object geometry set $\mathbb{V}$ and the grasp configuration set $\mathbb{G}$ are first passed through fully connected multi-layer perceptron (MLP)

heads in parallel, followed by bidirectional gated recurrent units (Bi-GRU) to capture sequential dependencies, and subsequently aggregated using mean operators. Their outputs are concatenated with the feature vector of the center grasp configuration G_{center} processed by another MLP head, and the feature vector of the environment map $\mathbb{M}$ processed by a ResNet18 encoder. The resulting concatenated feature vector is then further processed and normalized to produce the unit-length center embedding vector $\mathbf{e}_{\text{center}}$.

Context Embedding Network $f_{context}$: This network processes a subset of context grasp configurations $\mathbb{G}_{\text{context}} \subseteq \mathbb{G}$ and outputs their embeddings as a matrix $\mathbf{E}_{\text{context}} = f_{\text{context}}(\mathbb{G}_{\text{context}}, \Phi)$, where each row vector $\mathbf{e}_{\text{context}} \in \mathbb{R}^d$ represents the embedding of a grasp configuration $G_{\text{context}} \in \mathbb{G}_{\text{context}}$. The function f_{context} is implemented as a neural network, with its architecture shown in Fig. 4(b). The environment map $\mathbb{M}$, object geometry set $\mathbb{V}$, and full grasp configuration set $\mathbb{G}$ are processed in the same manner as in the center embedding network. Each element of $\mathbb{G}_{\text{context}}$ is first passed through a shared MLP head in parallel, producing individual feature vectors. These are concatenated with the feature representations of $\mathbb{M}$, $\mathbb{V}$, and $\mathbb{G}$ to form a set of larger feature vectors, which are then processed sequentially by a Bi-GRU, followed by an MLP and normalization layer. The final output is the matrix $\mathbf{E}_{\text{context}}$, whose rows are the unit-length context embedding vectors for $\mathbb{G}_{\text{context}}$.

The similarity between the embeddings $\mathbf{e}_{\text{center}}$ and $\mathbf{e}_{\text{context}}$ is used to estimate the conditional probability that a G_{context} is compatible with G_{center}:

$$\mathbb{P}(G_{\text{context}} \,|\, G_{\text{center}}) = \frac{\exp(\mathbf{e}_{\text{context}} \cdot \mathbf{e}_{\text{center}})}{\sum_{j=1}^{m} \exp(\mathbf{e}^j_{\text{context}} \cdot \mathbf{e}_{\text{center}})}, \quad (1)$$

where $\mathbf{e}_{\text{context}}$ is the embedding corresponding to G_{context}, $\mathbf{e}^j_{\text{context}}$ is the embedding corresponding to each $G^j \in \mathbb{G}$, and $m = |\mathbb{G}|$ is the total number of all grasp configurations.

B. Model Training

All model parameters are optimized during training, except for the weights of the ResNet18 backbone; only the final fully-connected head layers are fine-tuned, while the preceding convolutional blocks remain frozen. Layer normalization is applied after every hidden linear layer in the MLP-Embed and in the final MLPs that generate the center and context embeddings, providing regularization and stabilizing optimization. The CE model is trained by maximizing the following log-likelihood:

$$\sum_{G_{\text{center}} \in \mathbb{G}} \sum_{G_{\text{context}} \in \mathcal{DC}(G_{\text{center}})} \log \mathbb{P}(G_{\text{context}} \,|\, G_{\text{center}}), \quad (2)$$

where the conditional probability $\mathbb{P}(G_{\text{context}} \,|\, G_{\text{center}})$ is given by the softmax formulation in (1). The *Dynamic Context (DC)* set $\mathcal{DC}(G_{\text{center}}) \subseteq \mathbb{G}$ includes all context grasp configurations G_{context} that satisfy the feasibility condition $S(G_{\text{center}}, G_{\text{context}}) = 1$.[1]

Directly computing the softmax denominator in (1) becomes computationally expensive when the number of candidate configurations m is large. To mitigate this, we adopt a negative sampling-based training strategy inspired by the Word2Vec framework [19]–[21], which replaces the softmax training objective with a set of binary classification problems. Specifically, we introduce a binary random variable $D \in \{0, 1\}$ for each pair $(G_{\text{center}}, G_{\text{context}})$, where $D = 1$ if $G_{\text{context}} \in \mathcal{DC}(G_{\text{center}})$ and $D = 0$ otherwise. The conditional probability of this event is modeled using a sigmoid function applied to the dot product of the corresponding embedding vectors:

$$\mathbb{P}(D{=}1 | G_{\text{center}}, G_{\text{context}}) {=} \frac{1}{1{+}\exp(-\mathbf{e}_{\text{context}} \cdot \mathbf{e}_{\text{center}}/\tau)}, \quad (3)$$

where τ is a parameter that controls the sharpness of the sigmoid function.

Using this expression, the conditional probability in (1) is replaced by:

$$\mathbb{P}(D = 1 \mid G_{\text{center}}, G_{\text{context}}) \times \tfrac{1}{|\mathcal{N}(G_{\text{center}})|} \textstyle\prod_{G \in \mathcal{N}(G_{\text{center}})} (\mathbb{P}(D = 0 \mid G_{\text{center}}, G)), \quad (4)$$

where $\mathcal{N}(G_{\text{center}})$ is a set of negative samples—grasp configurations uniformly drawn from $\mathbb{G} \backslash \mathcal{DC}(G_{\text{center}})$. For each G_{center}, the number of negative samples is chosen such that the size of the union $\mathcal{DC}(G_{\text{center}}) \cup \mathcal{N}(G_{\text{center}})$ remains constant across all center grasp configurations.[2]

C. Affinity Matrix Computation and Grasp Selection

Once trained, the two neural networks are used to generate embeddings for all candidate grasp configurations in $\mathbb{G}$. The resulting embeddings are organized into two matrices: the center embedding matrix $\mathbf{E}_{\text{center}} \in \mathbb{R}^{m \times d}$ and the context embedding matrix $\mathbf{E}_{\text{context}} \in \mathbb{R}^{m \times d}$. The matrix $\mathbf{E}_{\text{center}}$ is formed by applying f_{center} to each configuration in $\mathbb{G}$. An affinity matrix is then computed as $A = \mathbf{E}_{\text{center}} \cdot \mathbf{E}^{\top}_{\text{context}}$, where each entry reflects the similarity between a center-context grasp pair. The top-k entries of A—those with the highest similarity scores—are selected as the most promising grasp configuration pairs for collective object transport. These selected pairs correspond to those that maximize the probability in (3).

D. Extension to Multi-Robot Object Transport

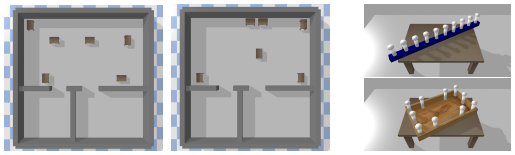
Recall that the CE model generates embeddings for all grasp configurations in $\mathbb{G}$, resulting in two sets: the center embeddings $(\mathbf{e}^1_{\text{center}}, \cdots, \mathbf{e}^m_{\text{center}})$ and context embeddings $(\mathbf{e}^1_{\text{context}}, \cdots, \mathbf{e}^m_{\text{context}})$. The model can be trained to maximize the following objective, extended to the N-robot setting:

$$\sum_{G_{\text{center}} \in \mathbb{G}} \sum_{P_N \in \mathcal{DC}(G_{\text{center}})} \log \mathbb{P}(P_N \,|\, G_{\text{center}}).$$

The *DC* set $\mathcal{DC}(G_{\text{center}}) \subset \mathbb{G}^{N-1}$ contains grasp configuration tuples $P_N = (G^{i_2}_{\text{context}}, \cdots, G^{i_N}_{\text{context}})$ for the remaining $N-1$ robots, where $\mathbb{G}^{N-1}$ denotes the $(N-1)$-fold Cartesian product of $\mathbb{G}$. A combination $(G_{\text{center}}, G^{i_2}_{\text{context}}, \cdots, G^{i_N}_{\text{context}})$ is

[1]The size of $\mathcal{DC}(G_{\text{center}})$ varies depending on G_{center}, and may contain many, few, or no feasible pairings.

[2]This reformulation decouples the complexity of evaluating (1) from the total number of grasp configurations $m = |\mathbb{G}|$, reducing it to scale with the number of negative samples, which can be chosen to be significantly smaller than m.



(a) Scenes (b) Training Objects

Fig. 5: Environments for collective object transport, featuring two table arrangements for object placement as illustrated in (a). The objects shown in (b) are used to generate the dataset for training the CE model, each with ten predefined grasp points for both the long bar and the rectangular object.

a feasible grasp configurations for N robots if and only if $P_N \in \mathcal{DC}(G_{\text{center}})$. The conditional probability can be defined using a softmax function as:

$$\mathbb{P}(P_N \mid G_{\text{center}}) = \frac{\exp(\frac{1}{N-1}(\mathbf{e}^{i_2}_{\text{context}} + \cdots + \mathbf{e}^{i_N}_{\text{context}}) \cdot \mathbf{e}_{\text{center}})}{\sum_{P'_N \in \mathbb{G}^{N-1}} \exp(\frac{1}{N-1}(\mathbf{e}^{i'_2}_{\text{context}} + \cdots + \mathbf{e}^{i'_N}_{\text{context}}) \cdot \mathbf{e}_{\text{center}})},$$

where $(\mathbf{e}^{i'_2}_{\text{context}}, \cdots, \mathbf{e}^{i'_N}_{\text{context}})$ are context embedding vectors corresponding to the grasp configurations in the tuple $P'_N = (G^{i'_2}_{\text{context}}, \cdots, G^{i'_N}_{\text{context}})$, such that $\mathbf{e}^{i'}_{\text{context}}$ is the embedding vector for $G^{i'}_{\text{context}}$.

To facilitate grasp selection across N robots, we define an affinity tensor $\mathcal{A} : \mathbb{G} \times \mathbb{G}^{N-1} \to \mathbb{R}$ as:

$$\mathcal{A}(G_{\text{center}}, P_N) = \tfrac{1}{N-1}(\mathbf{e}^{i_2}_{\text{context}} + \cdots + \mathbf{e}^{i_N}_{\text{context}}) \cdot \mathbf{e}_{\text{center}}.$$

Since $\mathcal{A}$ is computed as the dot product between the center embedding and the average of the context embeddings, it quantifies the collective compatibility of the grasp configurations. The optimal grasp configurations are then obtained by selecting the tuple $(G^*_{\text{center}}, P^*_N) = \arg\max_{(G_{\text{center}}, P_N)} \mathcal{A}(G_{\text{center}}, P_N)$.

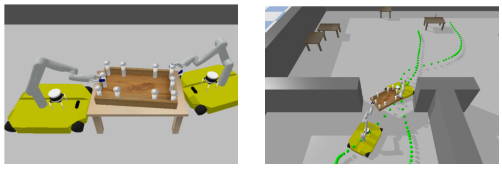
IV. Evaluation

A. Simulation Setup and Model Training

For model training, we employ two distinct table setups, as shown in Fig. 5(a), and two object types illustrated in Fig. 5(b). The first object is a long bar with white cylindrical grasping elements, while the second is a rectangular object with similar cylindrical elements distributed around its perimeter. Each cylindrical element serves a stable grasp point and corresponds to an element in the set $\mathbb{F}$.

The training dataset comprises 576 samples, generated by placing each object on one of six tables at 24 orientations uniformly sampled over the range $[-\pi, \pi)$. Each object is initially positioned on a table, and its destination is set as the center of the nearest unoccupied room on the opposite side of the corridor. After filtering out grasp configuration candidates that are infeasible for object grasping due to collisions with obstacles, the size of the grasp configuration set $\mathbb{G}$ typically ranges from 20 to 150.

For each sample, all feasible pairs of grasp configurations are identified based on their ability to successfully complete the object transport task, as determined by the trajectory planner (detailed in Appendix B) and illustrated in Fig. 6. The planner is implemented using Cyipopt [22] and PyBullet [23]. Dataset generation was performed on a high-performance computing cluster with 1000 Intel® Xeon® Gold 6148 CPUs, with each feasibility check taking under 30 seconds. [3]



(a) (b)

Fig. 6: (a) Validation of grasp configuration pairs for collective object transport in PyBullet. (b) Trajectory generation for a selected grasp configuration pair using the trajectory planner.

TABLE I: Hyperparameters for Training

Scheduler Parameters		**Model Architecture**	
LR-schedule factor	0.3741	Embed size (d)	44
Schedule patience	3 epochs		
Training Parameters		**Optimizer Parameters**	
Batch size	37	Weight decay	1×10^{-4}
Epochs	83	Learning rate	2.61×10^{-4}
Scaling factor (α)	1.10	Optimizer	AdamW
Sharpness (τ)	6.15×10^{-2}		

The learning objective in (2) is implemented as follows:

$$\sum_{G_{\text{center}} \in \mathbb{G}} \Big(\sum_{G_{\text{context}} \in \mathcal{DC}(G_{\text{center}})} \log p(G_{\text{center}}, G_{\text{context}}) + \frac{|\mathcal{DC}(G_{\text{center}})|}{|\mathcal{N}(G_{\text{center}})|} \sum_{G \in \mathcal{N}(G_{\text{center}})} \log(1 - p(G_{\text{center}}, G)) \Big), \quad (5)$$

where $p(G_{\text{center}}, G) = \mathbb{P}(D = 1 \mid G_{\text{center}}, G)$ defined as in (3). The negative set $\mathcal{N}(G_{\text{center}})$ is constructed to ensure that the union $\mathcal{DC}(G_{\text{center}}) \cup \mathcal{N}(G_{\text{center}})$ has a fixed size K across all G_{center} in $\mathbb{G}$. Specifically, $K = \alpha \max_{G_{\text{center}} \in \mathbb{G}} |\mathcal{DC}(G_{\text{center}})|$, where α is a hyperparameter that controls the ratio of feasible to infeasible samples and is tuned based on validation performance.

The full dataset is divided into a fixed set size of 70% for training, 20% for validation, and 10% for testing. Hyperparameters, summarized in Table I, are tuned using Bayesian optimization. The model is trained using the AdamW optimizer for 83 epochs on a single Nvidia RTX 3080 taking approximately 20 minutes.

B. Evaluation Results

Using the test dataset, we evaluate the CE model as a recommendation system by analyzing its top-k performance—ranking grasp configurations based on their scores in the affinity matrix derived from the model's embedding representations. Table II reports the success rate of identifying at least one feasible grasp configuration pair among the top-1, top-3, and top-5 ranked candidates. For reference, we also include a random selection policy to highlight the challenge of finding feasible pairs through random sampling. The model consistently achieves high success rates—surpassing 83% across all cases and reaching up to 99.22% for the top-5 candidates—compared to 41.11% achieved by random sampling.

We also evaluate the CE model's ability to distinguish between feasible and infeasible grasp configuration pairs using

[3] Our code is available at github.com/davidfalgo/cooperative_grasping.

TABLE II: Top-k Evaluation Results

Top-k Method	**Top-1** Success %	**Top-3** Success %	**Top-5** Success %
CE Model	83.05	96.39	99.22
Random	20.00	30.02	41.11

TABLE III: Evaluation of the CE Model's Discriminative Performance

Accuracy	Precision	Recall	F1-score	AUC-ROC
90.56%	92.02%	90.56%	90.91%	96.02%

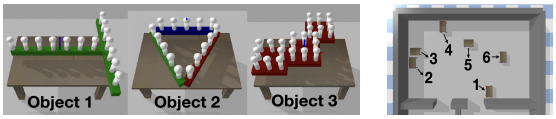
(3). Given the class imbalance—where infeasible pairs outnumber feasible ones—a threshold of 0.64 is applied to the output of (3), selected by maximizing the F1-score. Table III presents the evaluation results across various performance metrics. The reported *accuracy* of 90.56% indicates the proportion of correctly classified grasp configuration pairs. *Precision* measures the proportion of predicted feasible pairs that are actually feasible, while *recall* reflects the model's ability to identify all true feasible pairs. The high recall indicates that the model rarely misses feasible grasp pairs, and the high precision suggests a low false positive rate. The high *F1-score*—the harmonic mean of precision and recall—confirms balanced performance. Finally, the *AUC-ROC* score indicates the model assigns higher scores to feasible pairs than infeasible ones 96.02% of the time, confirming its strong discriminative capability.

C. Generalization to Novel Environments

Three new objects—each with a distinct shape and multiple grasp points, as illustrated in Fig. 7(a)—are specifically designed to assess the generalization capability of the CE model. The evaluation environment consists of six tables arranged as shown in Fig. 7(b). Each object is placed on one of the tables at ten uniformly spaced orientations over the range $[-\pi, \pi)$, resulting in 180 distinct evaluation scenarios. The number of candidate grasp configurations per scenario ranges from 80 to 450.

The CE model is evaluated based on its ability to recommend feasible grasp configuration pairs among the top-1, top-3, and top-5 ranked candidates. Table IV reports the overall success rates, including a random selection policy for reference. The results show that the proposed framework performs robustly even at small k, achieving a 91.11% success rate with top-5 predictions—demonstrating strong generalization to novel objects and unseen environments. Figure 8 further breaks down the success rates by object and table location, based on the top-5 selections. Among the test objects, Object 2 (triangular) achieved the maximum success rate at 98.3%, followed by Object 1 (T-shaped panel) at 91.7%, and Object 3 (asymmetric panel) at 86.7%. The average numbers of grasp configurations for Objects 1, 2, and 3 were 247, 297, and 267, respectively.

Performance also varied across table locations. Tables 5 and 6 achieved perfect success rates of 100%, with average numbers of valid grasp configurations at 359 and 367, respectively. Tables 1, 3, and 4 exhibited a slightly lower success rate of 96.7%, with an average of 296, 199, and 253 valid grasp configurations, respectively. In contrast, Table 2 had the lowest success rate at 63.3%, with only 149 valid grasp configurations on average. Most failures were concentrated around Table 2, where the object was tightly surrounded by obstacles (a wall and an adjacent table), severely restricting feasible grasp options. These results, as illustrated in Fig. 9, suggest a positive correlation between the success rate and the number of candidate grasp configurations.



(a) (b)

Fig. 7: (a) Three additional objects with diverse geometries used for generalization testing: Object 1, a T-shaped panel with 14 grasp points; Object 2, a triangular object with 15 grasp points; Object 3, an asymmetric panel with 24 grasp points. (b) Evaluation environment, with only the table-placed region shown.

TABLE IV: Top-k Evaluation Results for Assessing Model Generalization.

Top-k Method	**Top-1** Success %	**Top-3** Success %	**Top-5** Success %
CE Model	73.33	89.44	91.11
Random	18.89	33.33	45.56

In summary, object geometry and grasping flexibility are key factors influencing performance. Object 3 (asymmetric panel) exhibited the lowest success rate, despite having more graspable elements, due to its complex shape that complicates trajectory planning. In contrast, Object 1 (T-shaped panel) performed better, though failures were more common in scenarios with geometric occlusions that limited grasping options. Object 2 achieved an almost perfect success rate, benefiting from its symmetric design and well-distributed grasp points, which allowed robust performance even in constrained environments such as Tables 2, 3, and 4.

D. Evaluation on Physical Robot Platforms

We employed two *Clearpath Dingo-O* omnidirectional mobile bases, each mounted with a *Kinova Gen3-lite* manipulator. Robot and object poses were tracked using a motion capture system. The test environment, shown in Fig. 1, consists of two open areas connected by a narrow passage ($65\,\mathrm{cm} \times 50\,\mathrm{cm}$), allowing only one robot to pass at a time. An object is initially placed on a table in the first area, and the robots must transport it to a target location in the second area. We evaluated two object types: a straight bar ($63\,\mathrm{cm}$) and an L-shaped panel ($85\,\mathrm{cm} \times 60\,\mathrm{cm}$). In each trial, the CE model selected the grasp configuration with the highest affinity score, while the trajectory planner computed feasible paths for successful transport. These trajectories were then executed by each robot's controller for navigation. Figure 10 shows snapshots of the experiments: the top row depicts transport of the bar, and the bottom row shows the L-shaped object being carried through the passage.

V. Conclusions

This paper introduced a multi-robot framework for collective object transport centered on the CE model. This model

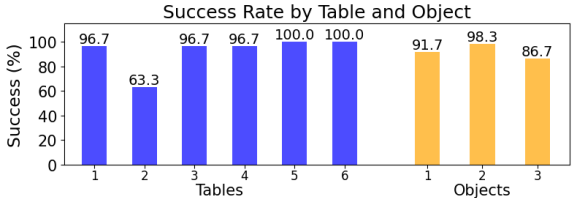


Fig. 8: Object transport success rates by tables and objects.

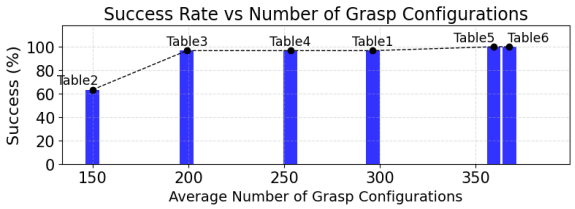


Fig. 9: Object transport success rates in relation to the number of candidate grasp configurations across different tables.

maps grasp configurations, object geometry, and environmental context into an embedding space to construct an affinity matrix, which identifies feasible grasp pairs for transport. Evaluation showed the framework reliably achieves high success rates and demonstrates strong generalization to novel objects, highlighting its practical utility. We plan to extend this research by incorporating object regrasping to improve success rates in scenarios with limited grasp configurations (e.g., moving the object to an open space to enable better candidates). Additionally, we aim to extend the framework to scenarios involving more than two robots.

APPENDIX

A. Sampling Grasp Configuration Candidates

To grasp an object, each robot must move to a base position (x_r, y_r) from which its end-effector can reach a designated grasp point (x_g, y_g) selected from the predefined set of grasp points $\mathbb{F}$ on the object. For each $(x_g, y_g) \in \mathbb{F}$, we generate candidate base positions (x_r, y_r) by uniformly sampling 60 points along a circle of radius r centered at (x_g, y_g). The sampling radius is set to $r = 0.55\,m$, which corresponds to an operationally effective reach distance from the center of the Dingo mobile base. This value falls within the robot arm's minimum and maximum reach limits ($0.35\,m$ and $0.7\,m$, respectively), ensuring that any sampled base position (x_r, y_r) allows the end-effector to reach the target grasp point (x_g, y_g) through forward extension.

All sampled positions are then filtered to retain only those from which the manipulator can reach the grasp point (x_g, y_g) without self-collision or collision with the tables and the object, and located within the free space $\mathbb{M}_{\text{free}}$. This process yields a set $\mathbb{G}$ of grasp configurations, where each configuration is represented as $G = (x_r, y_r, x_g, y_g)$.

B. Multi-Robot and Object Trajectory Planning

We formulate the trajectory planning problem for multiple robots and an object as an optimization problem. Using IRIS-NP [24], the robot's configuration space in $\mathbb{R}^2$ is decomposed into M overlapping convex polytopes $\{Q_r^{(1)}, \cdots, Q_r^{(M)}\}$ generated from M manually selected "seed" points located in the collision-free space $\mathbb{M}_{\text{free}}$. The polytopes are constructed such that $Q_r^{(1)}$ contains the initial positions of the two robots, $Q_r^{(M)}$ contains their destination, and each consecutive pair $Q_r^{(i)}$ and $Q_r^{(i+1)}$ has a nonempty intersection.

Similarly, the object's configuration space in $\mathbb{R}^2 \times [-\pi, \pi)$—representing its planar position and yaw orientation—is decomposed into a set of convex polytopes $\{Q_o^{(1)}, \cdots, Q_o^{(M)}\}$ using IRIS-NP. This decomposition leverages the same M seed positions used for the robot's configuration space, with each seed augmented by a carefully chosen orientation value from $[-\pi, \pi)$. The seeds are selected to lie entirely within the collision-free region of the object configuration space. The polytopes are constructed such that the following three requirements hold: $Q_o^{(1)}$ contains the object's initial position, $Q_o^{(M)}$ contains its destination, and each consecutive pair of polytopes, $Q_o^{(i)}$ and $Q_o^{(i+1)}$, is constructed to ensure a nonempty intersection.[4]

The set $Q = \bigcup_{k=1}^{M} Q^{(k)}$, where each region $Q^{(k)} = (Q_r^{(k)}, Q_o^{(k)})$, defines the joint feasible space of robot and object configurations. A trajectory $q : [0, T] \to Q$, comprising the positions of the two robots $\mathrm{x}_{r_1}(t), \mathrm{x}_{r_2}(t) \in \mathbb{R}^2$ and the position and yaw orientation of the object $\mathrm{x}_o(t) \in \mathbb{R}^2, \theta_o(t) \in [-\pi, \pi)$, is then planned to ensure collision-free motion while satisfying the kinematic constraints of both the robots and the object. Following the method in [25], we segment the trajectory $q(t)$, $t \in [0, T]$ based on its inclusion in regions $Q^{(k)}$, and parameterize each segment as a Bézier curve constrained to lie within its corresponding region $Q^{(k)}$. Each segment $q_k(t)$, $t \in [t_k, t_{k+1})$ is defined by four control points $\mathbf{c}_0^{(k)}, \cdots, \mathbf{c}_3^{(k)}$. Since resulting Bézier curves lie within the convex hull of their control points, ensuring $\mathbf{c}_0^{(k)}, \cdots, \mathbf{c}_3^{(k)} \in Q^{(k)}$ guarantees that the entire segment remains within the collision-free region. The control points $\mathbf{c}_0^{(k)}, \cdots, \mathbf{c}_3^{(k)} \in Q^{(k)}$, $k = 1, \cdots, M$ defining the entire trajectory from the initial state q_0 to the final state q_T are obtained by solving the following optimization problem:

$$
\begin{aligned}
\min \quad & L(\dot{q}, T) + V(\dot{q}, T) + S(\ddot{q}, T) + w_F F(\dot{q}, T) && \text{(6a)}\\
\text{s.t.} \quad & q(0) = q_0, \quad q(T) = q_T \\
& \dot{q}(0) = 0, \quad \dot{q}(T) = 0 \\
& ||\mathrm{x}_{r_i}(t) - P_G(t)\bar{\mathrm{x}}_{g_i}||_2 \geq r_{\min},\ t \in [0, T],\ i = 1, 2 && \text{(6b)}\\
& ||\mathrm{x}_{r_i}(t) - P_G(t)\bar{\mathrm{x}}_{g_i}||_2 \leq r_{\max},\ t \in [0, T],\ i = 1, 2 && \text{(6c)}\\
& ||\mathrm{x}_{r_i}(t) - P_G(t)\bar{\mathrm{x}}_{r_i}||_2 \leq r_{\text{collision},i},\ t \in [0, T],\ i = 1, 2 && \text{(6d)}\\
& \mathbf{c}_3^{(k)} - \mathbf{c}_2^{(k)} = \mathbf{c}_1^{(k+1)} - \mathbf{c}_0^{(k+1)}, && \text{(6e)}\\
& \mathbf{c}_3^{(k)} = \mathbf{c}_0^{(k+1)},\ k = 1, \cdots, M-1 && \text{(6f)}
\end{aligned}
$$

where $L(\dot{q}, T) = \int_0^T \|\dot{q}(t)\|_2 \,\mathrm{d}t$,
$V(\dot{q}, T) = \int_0^T \|\dot{q}(t)\|_2^2 \,\mathrm{d}t$, $S(\ddot{q}, T) = \int_0^T \|\ddot{q}(t)\|_2^2 \,\mathrm{d}t$,
$F(\dot{q}, T) = \sum_{i=1}^{2} \int_0^T \|\dot{\mathrm{x}}_{r_i}(t) - (\dot{\mathrm{x}}_o(t) + \dot{\theta}_o(t)\bar{\mathrm{x}}_{r_i}^{\perp})\|_2^2 \,\mathrm{d}t$.

The objective function in (6a) is composed of four terms: the trajectory length $L(q, T)$, the velocity penalty $V(\dot{q}, T)$, the smoothness term $S(\ddot{q}, T)$, and the formation consistency term

[4]The IRIS-NP algorithm requires *seed* points to generate convex polytopes. For this purpose, we followed the manual seeding process described in [25] to ensure that the resulting polytopes $\{Q_o^{(1)}, \cdots, Q_o^{(M)}\}$ satisfy the three requirements.

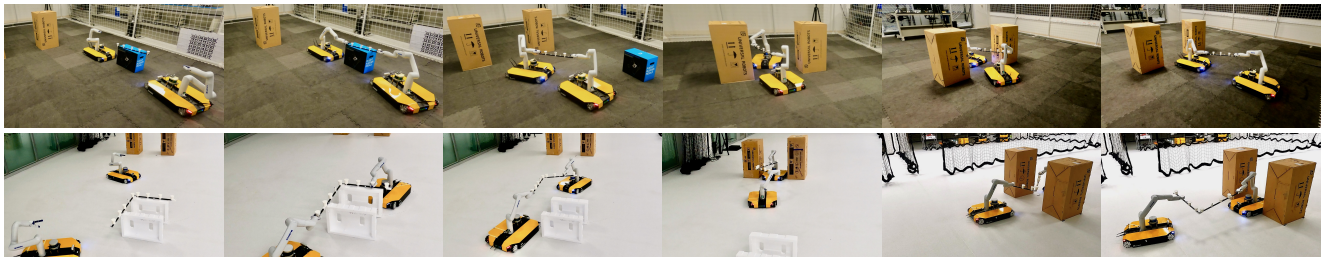


Fig. 10: Cooperative object transport using two *Clearpath Dingo-O* mobile manipulators, each equipped with a *Kinova Gen3-lite* arm. **Top row:** The robots transport a straight bar through a narrow corridor connecting two areas. **Bottom row:** The robots carry an L-shaped payload. Each Dingo-O platform measures $51\,\mathrm{cm}$ in width and $68\,\mathrm{cm}$ in length, while the corridor dimensions are $65\,\mathrm{cm}$ in width and $50\,\mathrm{cm}$ in length.

$F(\dot{q}, T)$. The last term $F(\dot{q}, T)$ measures the deviation between each robot i's velocity $\dot{\mathrm{x}}_{r_i}(t)$ and its expected rigid-body velocity, given by $\dot{\mathrm{x}}_o(t) + \dot{\theta}_o(t)\bar{\mathrm{x}}^{\perp}_{r_i}$. Here, $\bar{\mathrm{x}}^{\perp}_{r_i}$ is a perpendicular vector to the relative position $\bar{\mathrm{x}}_{r_i} = (\bar{x}_{r_i}, \bar{y}_{r_i})$, denoted as $\bar{\mathrm{x}}^{\perp}_{r_i} = (-\bar{y}_{r_i}, \bar{x}_{r_i})$. The relative position $\bar{\mathrm{x}}_{r_i}$ is the position of the i-th robot relative to the object's center and is chosen based on the selected grasp configuration G. This point is treated as rigidly attached to the object with translational velocity $\dot{\mathrm{x}}_o(t)$ and angular velocity $\dot{\theta}_o(t)$. This term penalizes deviations from ideal rigid-body behavior among the robots and the object. To control the influence of this term, we introduce a weight factor w_F (set to 20.0 in our implementation) that scales the contribution of $F(\dot{q}, T)$ in the overall objective.

To ensure feasible and coordinated manipulation, we impose three constraints. Firstly, each robot i must remain within a radial annulus defined by a minimum radius $r_{\text{min}} = 0.35$ and a maximum radius $r_{\text{max}} = 0.7$, centered at its grasp point $P_G(t)\bar{\mathrm{x}}_{g_i}$ at time t (constraints (6b), (6c)). Here, $P_G(t)$ is a time-varying transformation matrix that maps the grasp point $\bar{\mathrm{x}}_{g_i}$ from the object's local frame to the world frame at each time t. Secondly, the formation constraint (6d) ensures that each robot i remains within a distance $r_{\text{collision},i}$ of its nominal base position $P_G(t)\bar{\mathrm{x}}_{r_i}$, where $r_{\text{collision},i}$ is selected to as the maximum distance that guarantees collision-free operation between each robot i and the object, given a grasp configuration. Finally, continuity of the trajectory is enforced by ensuring $q(t)$ is continuously differentiable, as specified in constraints (6e), (6f).

REFERENCES

[1] G. Eoh, J. D. Jeon, J. S. Choi, and B. H. Lee, "Multi-robot cooperative formation for overweight object transportation," in *IEEE/SICE International Symposium on System Integration*, pp. 726–731, Dec 2011.

[2] N. Vahrenkamp, E. Kuhn, T. Asfour, and R. Dillmann, "Planning multi-robot grasping motions," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 593–600, Dec 2010.

[3] U. Tariq, R. Muthusamy, and V. Kyrki, "Grasp planning for load sharing in collaborative manipulation," in *2018 IEEE International Conference on Robotics and Automation*, pp. 6847–6854, May 2018.

[4] R. Muthusamy and V. Kyrki, "Decentralized approaches for cooperative grasp planning," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 693–698, Dec 2014.

[5] R. Muthusamy, C. P. Bechlioulis, K. J. Kyriakopoulos, and V. Kyrki, "Task specific cooperative grasp planning for decentralized multi-robot systems," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6066–6073, May 2015.

[6] R. Muthusamy, V. Kyrki, P. K. Muthusamy, T. Taha, I. Hussain, Y. Zweiri, D. Prattichizzo, and D. Gan, "Strictly decentralized approaches for multi-robot grasp coordination," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pp. 1–8, Aug 2023.

[7] W. Liu, M. Ren, K. Song, M. Y. Wang, and Z. Xiong, "A novel planning framework for complex flipping manipulation of multiple mobile manipulators," 2024.

[8] O. Nachum, M. Ahn, H. Ponte, S. S. Gu, and V. Kumar, "Multi-agent manipulation via locomotion using hierarchical sim2real," in *Proceedings of the Conference on Robot Learning* (L. P. Kaelbling, D. Kragic, and K. Sugiura, eds.), vol. 100 of *Proceedings of Machine Learning Research*, pp. 110–121, 30 Oct–01 Nov 2020.

[9] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, 2015.

[10] G. Habibi, Z. Kingston, W. Xie, M. Jellins, and J. McLurkin, "Distributed centroid estimation and motion controllers for collective transport by multi-robot systems," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1282–1288, 2015.

[11] H. Farivarnejad, S. Wilson, and S. Berman, "Decentralized sliding mode control for autonomous collective transport by multi-robot systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1826–1833, 2016.

[12] H. J. Savino, L. C. Pimenta, J. A. Shah, and B. V. Adorno, "Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators," *Journal of the Franklin Institute*, vol. 357, no. 1, pp. 142–178, 2020.

[13] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[14] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6466–6472, 2021.

[15] P. Vlantis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Multi-robot cooperative object transportation with guaranteed safety and convergence in planar obstacle cluttered workspaces via configuration space decomposition," *Robotics*, vol. 11, no. 6, 2022.

[16] F. Kennel-Maushart and S. Coros, "Payload-aware trajectory optimisation for non-holonomic mobile multi-robot manipulation with tip-over avoidance," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 7669–7676, 2024.

[17] G. Eoh and T.-H. Park, "Cooperative object transportation using curriculum-based deep reinforcement learning," *Sensors*, vol. 21, no. 14, 2021.

[18] L. Zhang, Y. Sun, A. Barth, and O. Ma, "Decentralized control of multi-robot system in cooperative object transportation using deep reinforcement learning," *IEEE Access*, vol. 8, 2020.

[19] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge University Press, 2023. https://D2L.ai.

[20] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.

[22] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[23] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning." http://pybullet.org.

[24] M. Petersen and R. Tedrake, "Growing convex collision-free regions in configuration space using nonlinear programming," *CoRR*, vol. abs/2303.14737, 2023.

[25] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Sci. Robotics*, vol. 8, no. 84, 2023.